

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI  
I TECHNIK INFORMACYJNYCH



Instytut Telekomunikacji

# Praca dyplomowa magisterska

na kierunku telekomunikacja  
w specjalności telekomunikacja

Rekonesans DNS na podstawie analizy zawartości zapytań AXFR

Marcin Skwarek

Numer albumu 257319

promotor  
dr hab. inż Wojciech Mazurczyk

WARSZAWA 2017



# **Rekonesans DNS na podstawie analizy zawartości zapytań AXFR**

## **Streszczenie**

Celem niniejszej pracy jest omówienie konsekwencji lekceważenia zabezpieczania protokołu DNS, które może powodować poważne incydenty bezpieczeństwa sieciowego. Protokół Domain Name System jest jednym z kluczowych elementów sieci Internet, bez którego niemożliwe byłoby jej działanie. Z racji na swoją istotną rolę, powinien być on jak najlepiej zabezpieczony przed różnego rodzaju atakami oraz być niezawodnym. Niezawodność została zapewniona między innymi poprzez wprowadzenie redundancji serwerów przestrzeni nazw. Bardzo ważne jest również zapewnienie, że informacje przesyłane protokołem DNS są spójne oraz autentyczne. Należy także pamiętać o zachowaniu ich prywatności. Informacje na temat infrastruktury oraz usług przechowywane są w strefach DNS. System DNS został usprawniony dzięki wprowadzeniu mechanizmu AXFR, który umożliwia transfer strefy z jednego serwera na drugi. Wynikiem transferu tego typu, jest otrzymanie wpisów przechowywanych w danej strefie. Nieumiejenna konfiguracja serwera DNS może prowadzić do tego, że każda maszyna będzie mogła skutecznie pobrać strefę DNS danego serwera. Nieumyślne ujawnianie pozornie nieszkodliwych informacji, może prowadzić do poważnych problemów związanych z bezpieczeństwem sieciowym. W pracy przeanalizowane zostały różnice oraz podobieństwa serwerów, które umożliwiają przeprowadzenie takiego transferu. Ponadto, przedstawione zostały korzyści, które może czerpać cyberprzestępca w wyniku pozyskania informacji o strefach DNS. Cel został osiągnięty poprzez realizację globalnego skanowania domen pod kątem transferu AXFR oraz późniejszej syntezy wyników.

**Słowa kluczowe:** *DNS, rekonesans, cyberbezpieczeństwo, AXFR*



# **DNS reconnaissance based on content of AXFR queries analysis**

## **Abstract**

The purpose of the thesis is to present consequences of disregard for DNS data protection. Domain Name System protocol is one of crucial elements on the Internet that run network. Due to its significant role, it has to be properly secured against various attacks and reliable. The latter is achieved through implementation of redundant namespace server. Furthermore, information sent through DNS should be coherent and authentic, keeping in mind its privacy. Infrastructure and services information is stored in DNS zones. DNS was streamlined by introducing AXFR protocol that defines zone transfer from one server to another. The result of that type of transfer is obtaining entry in certain zone. Improper DNS server configuration can lead to the situation that any computer will be able to download zone files successfully. Unintentional disclosure of potentially harmless information may lead to severe network security problems. Thesis involves analysis of differences and similarities between servers, which enables carrying out transfer operations. There are also presented potential benefits for cybercriminal gaining information about DNS zones. The aim may be achieved by global domain scanning, paying special attention to AXFR transfer and further synthesis of results.

**Keywords:** *DNS, reconnaissance, cybersecurity, AXFR*





*Maciej Skwerek*

imię i nazwisko studenta  
name and surname of the student  
*257319*

numer albumu  
student record book number  
*telekomunikacja*

kierunek studiów  
field of study

*Warszawa 11.09.2012*

miejscowość i data  
place and date

## OŚWIADCZENIE

### DECLARATION

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

*Under the penalty of perjury, I hereby certify that I wrote my diploma thesis on my own, under the guidance of the thesis supervisor.*

Jednocześnie oświadczam, że:

*I also declare that:*

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- *this diploma thesis does not constitute infringement of copyright following the act of 4 February 1994 on copyright and related rights (Journal of Acts of 2006 no. 90, item 631 with further amendments) or personal rights protected under the civil law,*
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- *the diploma thesis does not contain data or information acquired in an illegal way,*
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem dyplomów lub tytułów zawodowych,
- *the diploma thesis has never been the basis of any other official proceedings leading to the award of diplomas or professional degrees,*
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- *all information included in the diploma thesis, derived from printed and electronic sources, has been documented with relevant references in the literature section,*
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.
- *I am aware of the regulations at Warsaw University of Technology on management of copyright and related rights, industrial property rights and commercialisation.*



**Politechnika Warszawska**  
Warsaw University of Technology

załącznik nr 10 do zarządzenia  
nr /2016 Rektora PW

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

*I certify that the content of the printed version of the diploma thesis, the content of the electronic version of the diploma thesis (on a CD) and the content of the diploma thesis in the Archive of Diploma Theses (APD module) of the USOS system are identical.*

*Marcin Słowiak*

czytelny podpis studenta  
*legible signature of the student*

# Spis treści

<b>1 Wstęp</b>	<b>7</b>
1.1 Ataki rekonesansowe . . . . .	7
1.2 Domain Name System . . . . .	10
1.2.1 Podział domen najwyższego poziomu . . . . .	12
1.2.2 Domena DNS . . . . .	13
1.2.3 Strefa DNS . . . . .	13
1.2.4 Strefa DNS a domena DNS . . . . .	13
1.2.5 Kompletne nazwy domen . . . . .	14
1.2.6 Mapowanie nazw na adres IP . . . . .	14
1.2.7 Mapowanie odwrotne . . . . .	15
1.2.8 Wpisy plików strefy . . . . .	15
1.2.9 TLD ARPA . . . . .	18
1.2.10 Plik strefy DNS . . . . .	19
1.3 Transfer strefy DNS . . . . .	20
1.3.1 Transfer AXFR . . . . .	21
1.3.2 Transfer IXFR . . . . .	21
1.4 Podpisy TSIG . . . . .	22
1.4.1 Opis działania TSIG . . . . .	22
1.4.2 Wady TSIG . . . . .	23
1.5 Inne metody podpisywania wiadomości . . . . .	23
1.6 Parkowanie domen . . . . .	24
<b>2 Powiązane prace</b>	<b>27</b>
2.1 Internet-Wide Scan Data Repository . . . . .	27
2.2 Projekty open source . . . . .	28
2.3 DNS Response Policy Zone . . . . .	29
2.4 DNS as a service . . . . .	29
2.5 DNS Enumeration . . . . .	30

<b>3</b>	<b>Implementacja</b>	<b>35</b>
3.1	Przegląd dostępnych narzędzi . . . . .	35
3.2	Zaimplementowane narzędzia . . . . .	38
3.2.1	Program dig zarządzany skryptami powłoki . . . . .	39
3.2.2	Menedżer procesów zarządzany skryptami języka Python . . . . .	39
3.2.3	Skaner C-AXFR . . . . .	40
3.3	Dane wejściowe . . . . .	44
3.4	Środowisko uruchomieniowe . . . . .	45
3.5	Metodyka badań . . . . .	46
3.6	System powiadomień . . . . .	47
<b>4</b>	<b>Badania eksperymentalne i uzyskane wyniki</b>	<b>49</b>
4.1	Typy odebranych odpowiedzi . . . . .	49
4.2	Odebrane adresy IPv4 oraz IPv6 . . . . .	51
4.2.1	Analiza AS . . . . .	51
4.2.2	Sposób określenia AS . . . . .	52
4.2.3	Zbiór adresów IPv4 . . . . .	53
4.2.4	Zbiór adresów IPv6 . . . . .	55
4.3	Analiza TLD . . . . .	56
4.4	Liczba wpisów stref w odpowiedziach . . . . .	61
4.5	Odpowiedzi niestandardowe . . . . .	65
4.6	Geograficzna lokalizacja serwerów . . . . .	68
4.7	Usługi . . . . .	70
4.7.1	Usługi wspomagające rozwój oprogramowania . . . . .	71
4.7.2	Usługi poczty elektronicznej . . . . .	71
4.8	Strefy DNSSEC . . . . .	72
4.9	Wpisy SPF . . . . .	73
4.10	Strefy podsieci . . . . .	74
<b>5</b>	<b>Podsumowanie</b>	<b>77</b>
5.1	Zalecenia . . . . .	77
5.2	Retrospekcja . . . . .	78

# **1. Wstęp**

Ataki na systemy informatyczne czy telekomunikacyjne są bardzo złożone. Wymagają one bardzo często ciężkiej pracy już przed właściwym przejęciem kontroli nad systemem. Pozyskuje się wtedy informacje, których posiadanie może mieć istotny wpływ w późniejszych etapach ataku. Skuteczny atak na system powinien przebiegać w jak najkrótszym czasie, tak, aby dać jak najmniej czasu drugiej stronie na jego odparcie. Właśnie dlatego cyberprzestępcy starają się pozyskiwać jak największej informacji o celach ataków. Informacje odnoszą się do wielu płaszczyzn – zarówno technicznych aspektów jak i socjologicznych bądź społecznych.

Celem niniejszej pracy magisterskiej było zbadanie jakie informacje możliwe są do uzyskania na temat systemów teleinformatycznych bazując na protokole Domain Name System. Osiągnięcie tego celu zapewniono poprzez wykonanie skanowania podatności domen na transfer strefy oraz późniejszą syntezę pobranych informacji. Podejście zaprezentowane w niniejszej pracy magisterskiej jest unikalne na światową skalę ze względu na to, jak duży obszar został pokryty podczas wykonywania skanowania.

W pracy magisterskiej zaprezentowano oraz przedyskutowano dostępność i jakość narzędzi umożliwiających przeprowadzenie takich badań. Następnie zaproponowano autorskie podejście do implementacji narzędzi umożliwiających realizację prac na taką skalę oraz opisano architekturę środowiska uruchomieniowego. W kolejnej części pracy przedstawiono przebieg badań eksperymentalnych oraz uzyskane wyniki, które zostały następnie poddane analizie i syntezy. Przeanalizowano, jakimi cechami wspólnymi charakteryzują się uzyskane odpowiedzi oraz zaprezentowano i omówiono anomalie, które wystąpiły podczas badań eksperymentalnych. Odniesiono się do najbardziej użytecznych informacji z punktu widzenia cyberprzestępców, tj. adresów IP oraz usług, uruchomionych pod danymi adresami. W podsumowaniu pracy uwydatnione zostały najbardziej istotne aspekty wiążące się z prezentowaną tematyką oraz wydano zalecenia umożliwiające poprawę bezpieczeństwa obszaru odnoszącego się do protokołu DNS, które kierowane są do administratorów domen.

## **1.1 Ataki rekonesansowe**

Ataki rekonesansowe są typem ataków komputerowych, których głównym celem jest pozyskanie informacji na temat atakowanego systemu bądź podatności, które w nim występują. Słowo „re-

rekonesans” zostało zapożyczone z nomenklatury militarnej i odnosi się do zapoznania z terenem wroga. W kontekście ataków na sieci komputerowe stwierdzeniem określa się analogiczny krok – działanie przed właściwym atakiem. Sam rekonesans można podzielić dodatkowo na dwie kategorie: atak aktywny oraz pasywny. Atak aktywny odnosi się do działania, gdy atakujący podejmuje akcje, przez które może wchodzić w interakcję z systemem, na przykład wysyłanie specjalnie spersonowanych zapytań czy skanowanie portów.

Atak pasywny to tylko i wyłącznie obserwacje działającego systemu. Może to być na przykład podsłuchiwanie ruchu, analiza najczęściej odwiedzanych stron, czy choćby przyglądarki się innym procesom aby odpowiednio przygotować atak właściwy.

Obie wersje ataków rekonesansowych są również częścią tak zwanego etycznego hakowania (ang. *ethical hacking*). Osoby które tym się zajmują (określani po angielsku jako *white hat*) starają się wytknąć błędy i podatności w systemach przy czym starają się nie ingerować w ich działanie.

Innym podziałem, który pojawia się w kontekście ataków rekonesansowych, jest podział na 4 grupy. Każda z nich określa atak przeprowadzany w innych warunkach, środowisku. Zaproponowany podział uwzględnia nie tylko techniczne aspekty pozyskiwania informacji o systemie, ale również kwestie społeczne, jak na przykład próby uzyskania danych od pracowników firm, bądź ogólnie użytkowników systemów. Kompletną listę grup wyróżnionych w tym podziale zaprezentowano oraz krótko opisano w tabeli 1.1.

Lp.	Obszar pozyskiwania informacji	Opis
1	Profil sieci	Pozyskiwanie informacji na temat infrastruktury sieciowej: adresy IP, domeny, poddomeny; informacje na temat topologii systemu.
2	Profil środowiska	Pozyskiwanie informacji na temat przypadków użycia systemu, aktorów, grup, nazw użytkowników, wykorzystanego sprzętu, architektury, portów na których uruchomione są usługi.
3	Profil użytkowników systemu	Informacje „społeczne” – imiona, nazwiska, numery telefonów, plotki, określenie poziomu wiedzy na temat systemów telekomunikacyjnych, informatycznych, zabezpieczeń.
4	Profil zabezpieczeń	Określenie poziomu, profilu zabezpieczeń fizycznych (np. kontroli dostępu), złożoności haseł wymaganych przez system, pozyskanie informacji na temat systemów wykrywania włamania.

Tabela 1.1: Typy rekonesansu, opis na podstawie [41].

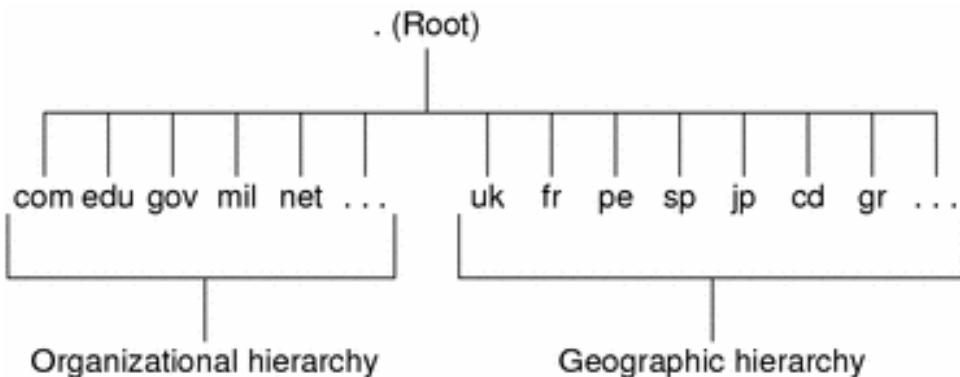
Z uwagi na techniczny charakter niniejszej pracy magisterskiej w kolejnych rozdziałach skupiono się głównie na rekonesansie odnoszącym się do infrastruktury systemów. Mowa tu zarówno o infrastrukturze sieciowej jak i profilu sprzętowym odpytywanych maszyn. Ataki tego typu są najczęściej przeprowadzane poprzez wykonanie kilku testów, jednak nie są to jedyne sposoby pozyskiwania informacji o infrastrukturze. Jednym z podstawowych, powszechnie spotykanych przykładów aktywnego rekonesansu jest skanowanie portów. Proces ten jest zaliczany do ataku aktywnego, ponieważ atakujący próbuje dowiedzieć się, na których portach danego systemu uruchomione są usługi (np. FTP, www, serwer pocztowy, SSH). Uzyskanie takich danych pozwala w późniejszym etapie na wykorzystanie podatności danych usług, wynikających głównie z błędnej implementacji bądź konfiguracji. Prowadzi to następnie do eskalacji uprawnień na maszynie na rzecz atakującego. Przykładowe wykorzystanie danych podatności na podstawie usługi SSH opisane zostało w dokumencie [25].

Inną formą rekonesansu dotyczącego infrastruktury systemu jest rekonesans DNS. Jest to część testu penetracyjnego polegającą na pozyskaniu jak największej ilości informacji na temat badanej domeny. Dane uzyskiwane podczas jego przeprowadzania odnoszą się zarówno do serwera DNS jak i wpisów, które przechowuje. Zebrane informacje mogą kompromitować infrastrukturę sieciową firmy nie powodując przy tym generowania zbyt podejrzanego ruchu. Między innymi dlatego ważne jest, aby przywiązywać znaczną uwagę do tego kto i w jaki sposób próbuje łączyć się z serwerami autorytywnymi odpowiedzialnymi za domenę. W sytuacji, gdy badana jest jedna domena, bądź zbiór kilku domen pod zarządem jednej organizacji wykorzystywane są najczęściej narzędzia dostępne w różnego rodzaju pakietach, na przykład BIND [50]. Przykładami takich programów są dig [51] czy nslookup [72]. Oba pozwalają na wysyłanie zapytań DNS do serwerów. Największą zaletą wspomnianych narzędzi jest możliwość dostosowywania pól w wiadomościach do potrzeb użytkownika. Dodatkowo nslookup umożliwia również pracę w trybie serwera, jednak bardziej istotna dla atakujących jest sama generacja zapytań, dzięki którym można uzyskać informację na temat testowanego systemu.

Jednym z największych błędów, które można popełnić przy konfiguracji serwera DNS jest umożliwienie przeprowadzenia transferu strefy DNS przez nieautoryzowane serwery. Jest to istotna kwestia dla każdego z serwerów do których można kierować zapytania z sieci zewnętrznej. W takim przypadku atakujący otrzymuje informacje o całej strefie przechowywanej na serwerze. Problem ten staje się jeszcze bardziej poważny w momencie, gdy serwer DNS obsługuje zarówno sieć wewnętrzna jak i zewnętrzną. Ujawnienie danych systemu DNS sieci wewnętrznej jest równoznaczny z udostępnieniem planu sieci, która jest przez niego obsługiwana.

AXFR (*ang. Asynchronous Xfer Full Range*) [66, 67] to mechanizm używany w protokole DNS (*Domain Name System*) do transferowania strefy, za którą odpowiada serwer nazw. Głównym przeznaczeniem opisywanego standardu był transfer informacji pomiędzy podstawowym i zapasowym serwerem przestrzeni nazw. Zasada jego działania jest bardzo prosta – serwer podrzędny (*ang. slave*) przesyła żądanie AXFR do serwera podstawowego (*ang. primary, master*).

Oczywiście jest, że AXFR jest wykorzystywany w celach zupełnie innych niż te, do których go



Rysunek 1.1: Hierarchia systemu DNS [71].

zaprojektowano. Mowa tu o sytuacji, w której serwer główny w żaden sposób nie weryfikuje po swojej stronie źródła takiego zapytania. Prowadzi to do sytuacji, w której każdy, kto jest w stanie utworzyć odpowiedni pakiet TCP może wejść w posiadanie informacji o całej strefie, za którą odpowiada odpytywany serwer DNS. Wspomniane przygotowanie pakietu DNS nie jest specjalnie trudne, ponieważ umożliwia to wiele narzędzi, na przykład dig [51], wchodzący w skład pakietu bind [50].

## 1.2 Domain Name System

Głównym zadaniem protokołu DNS (*Domain Name System*) [67] jest translacja nazw przyswajalnych dla użytkowników (najczęściej alfanumerycznych) na nazwy sieciowe, czyli adresy IP.

DNS jest jednym z podstawowych elementów internetu. Z wystawionego przez niego interfejsu korzysta wiele usług sieciowych i innych protokołów. Można traktować go jako bazę danych, która jest rozproszona po wielu lokalizacjach. Ponadto, system powinien oraz cechuje się dużą niezawodnością. W tym przypadku została ona osiągnięta poprzez wprowadzenie dość prostego mechanizmu – nadmiarowości serwerów. To właśnie dzięki tej redundancji DNS cechuje się wysokim wskaźnikiem niezawodności, gdyż w momencie gdy któryś z serwerów nie odpowiada informacja pobierana jest z innego serwera – zapasowego, podrzędnego.

System DNS ma charakterystyczną, hierarchiczną strukturę, którą zaprezentowano na rysunku 1.1. Na rysunku przedstawiono węzeł główny (ang. *root*) reprezentowany jako znak pojedynczej kropki oraz przykład dwóch typów domen pierwszego poziomu.

Dzięki tej hierarchii systemu możemy powiedzieć o DNS, że charakteryzuje go bardzo dobra skalowalność oraz elastyczność.

Podział ze względu na położenie geograficzne jest oczywiście bardziej naturalny i łatwy zarówno do wdrożenia jak i zrozumienia. Każdemu z państw przydzielono dwuliterowy identyfikator,

który reprezentuje domenę najwyższego poziomu odpowiadającą danemu państwu. Dozwolone są także trzyliterowe identyfikatory domen najwyższego poziomu (np com, net, org), które z reguły przypisywane są różnego rodzaju organizacjom. Powołując się na informacje przedstawione na grafice 1.1 TLD o identyfikatorze *uk* odpowiada domenom utożsamianym z Wielką Brytanią, a *fr* – domenom francuskim.

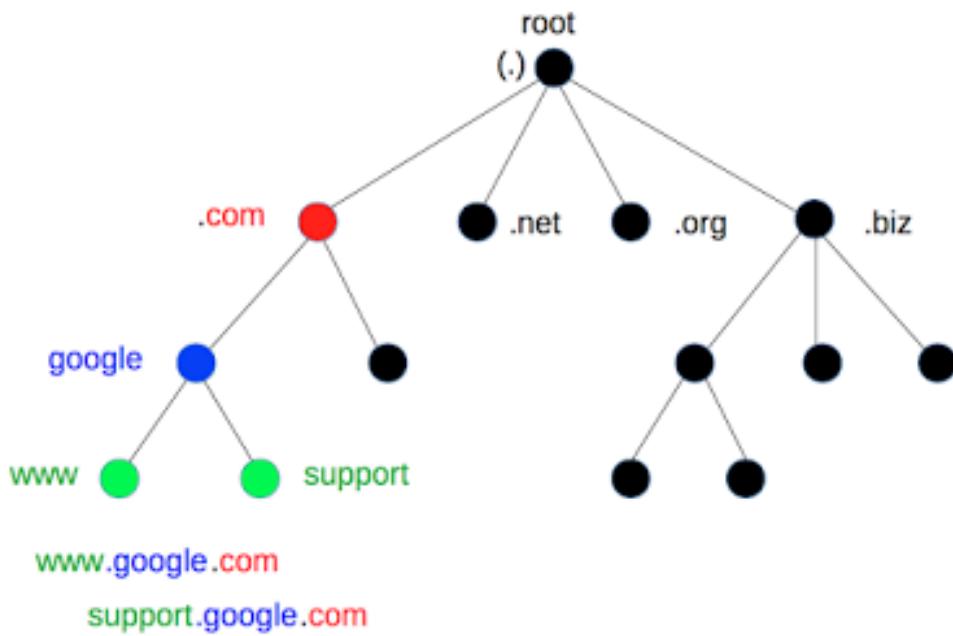
Hierarchia systemu DNS wynika z faktu, że domenami internetowymi każdego poziomu może zarządzać inna organizacja. Oznosi się to zarówno do domeny *root*, jak i domen najwyższego poziomu niezależnie od przynależności grupowej. W Polsce identyfikatorem TLD jest sufiks *pl*, a jednostką odpowiedzialną za nią jest CERT Polska [68]. Jeśli użytkownik chciałby dołączyć ze swoją siecią do internetu powinien zgłosić do odpowiedniej organizacji taką chęć oraz dostarczyć wszystkich niezbędnych informacji, które są wymagane przez zarządcę.

Bardzo ważnym punktem jest wspomniana w poprzednim akapicie „częć” dołączenia do internetu. Jeśli użytkownik czy organizacja chcą używać protokołu DNS jedynie do użytku wewnętrznego, to nie ma restrykcyjnych ograniczeń co do używanych nazw. Jeśli natomiast oczekuje się wystawienia domen w taki sposób, aby były widoczne z zewnątrz, to należy odpowiednio:

1. zarejestrować nazwę domeny,
2. pozyskać adres IP.

Bardzo trafne jest tu porównanie całego systemu Domain Name System do struktury plików w systemach operacyjnych z rodziny UNIX. Nazwa domeny bezpośrednio określa jej miejsce w całej przestrzeni nazw, podobnie jak ścieżka bezwzględna pliku określa jego miejsce w całym systemie plików. Po rejestracji domeny, jest ona dołączana w odpowiednie miejsce w hierarchii. Przykład domeny *google.com* razem z jej poddomenami i odpowiednimi miejscami w hierarchii systemu przedstawiono na rysunku 1.2.

Serwer autorytywny (ang. *authoritative name server*) posiada odgórne przyzwolenie na zarządzanie nazwami swoich hostów. Wymagania dotyczące technicznych kwestii serwerów autorytywnych zaprezentowano na stronie organizacji IANA [16], która w znacznym stopniu zajmuje się definiowaniem zasad obowiązujących w sieci Internet. Ze względu na drzewiastą strukturę systemu można oczekiwać, że kolejne domeny oraz ich serwery autorytywne będą delegować odpowiedzialność za kolejne strefy do serwerów niższego poziomu. W ten sposób, powołując się na przykład przedstawiony na rysunku 1.2 serwer autorytywny *.com* zarządza nazwami w domenie *.com*, natomiast zarządzanie nazwami w domenie *google.com* przekazuje do niższego szczeblem serwera przestrzeni nazw. W ten sposób serwer domeny *google.com* ma możliwość przypisywania nazw takim domenom jak zaprezentowane *support.google.com*.



Rysunek 1.2: Położenie domeny w przestrzeni nazw [48].

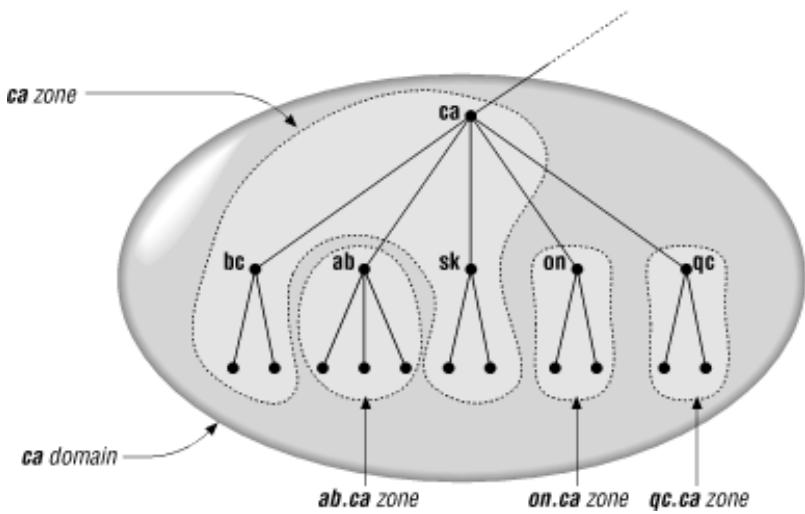
### 1.2.1 Podział domen najwyższego poziomu

Jeśli chodzi o podział domen pierwszego poziomu ze względu na przynależność organizacyjną, to aktualny stan przedstawiony jest w tabeli poniżej.

TLD	Opis jednostki
com	Jednostki o działalności komercyjnej (ang. <i>commercial institutions</i> )
edu	Jednostki edukacyjne (ang. <i>educational institutions</i> )
gov	Instytucje rządowe (ang. <i>government institutions</i> )
mil	Grupy wojskowe (ang. <i>military groupos</i> )
net	grupy związane z działaniem sieci (ang. <i>network support centers</i> )
org	Organizacje nonprofit i inne (ang. <i>nonprofit organizations</i> )
int	Organizacje międzynarodowe (ang. <i>international organizations</i> )

Tabela 1.2: Typy rekonesansu, opis na podstawie [41].

Przedstawiony podział nie jest stały. Autorzy zastrzegli, że w przyszłości może być on rozszerzony o dodatkowe kategorie.



Rysunek 1.3: Zakres drzewa hierarchii DNS wchodzący w skład domeny [72].

### 1.2.2 Domena DNS

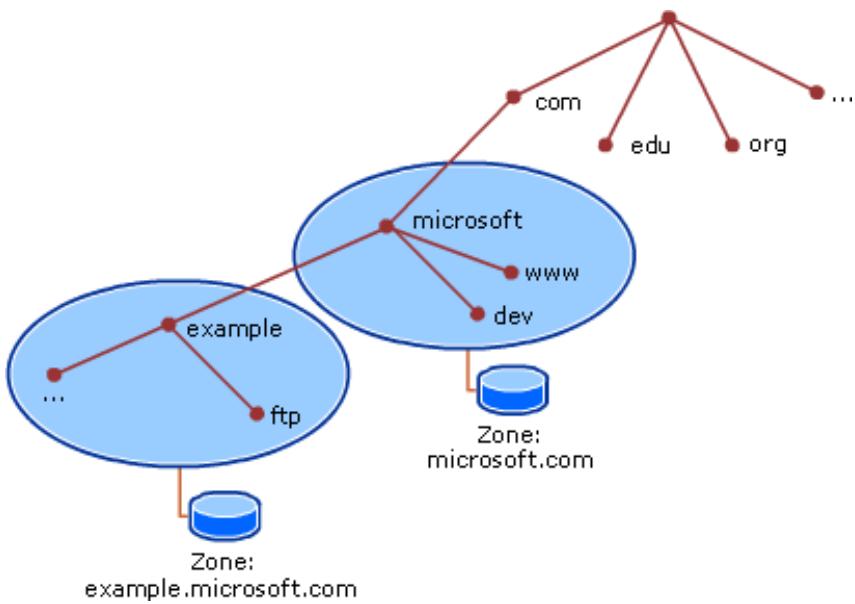
Domeną określa się dany podzbiór hierarchii DNS. Są to wszystkie poddomeny podlegające tej samej domenie wyższego poziomu. Odnosząc się do wcześniej przywołanego porównania do systemu plików – domena to odpowiednik folderu. Może zawierać kolejne domeny, może być określana zarówno na bardzo wysokim (domeny poziomu TLD) jak i bardzo szczegółowym (domeny 2-3 LD) poziomie abstrakcji. Jeśli chcielibyśmy reprezentować system DNS jako drzewo, to domeną DNS nazwiemy węzeł drzewa i wszystkie węzły, które są jego potomkami. Pojęcie domeny i elementy, które ono określa przedstawiono na rysunku 1.3.

### 1.2.3 Strefa DNS

Strefa DNS jest pojęciem, które określa zbiór domen, za które odpowiedzialny jest konkretny serwer przestrzeni nazw. Jeśli wyobrażymy sobie hierarchię DNS jako drzewo, w którym węzłami są kolejne nazwy domen, to strefą DNS będzie zbiór węzłów tego drzewa. Zbiór węzłów wyznaczany jest na podstawie informacji o serwerze autorytatywnym odpowiedzialnym za daną domenę. Te węzły (domeny) których nazwami zarządza ten sam serwer znajdują się w jednej strefie DNS. Przykładowy podział systemu na konkretne strefy został zaprezentowany na rysunku 1.4.

### 1.2.4 Strefa DNS a domena DNS

System DNS danej domeny jest zrealizowany w oparciu o zbiór serwerów przestrzeni nazw (ang. *nameserver*). Każdy z serwerów może być serwerem autorytatywnym dla pojedynczej domeny,



Rysunek 1.4: Zakres drzewa DNS określany terminem strefy DNS [64].

wielu domen bądź domen wraz z odpowiadającymi im poddomenami. Wycinek przestrzeni zarządzany przez określony serwer nazywany jest strefą DNS.

### 1.2.5 Kompletne nazwy domen

Określeniem pełne, kompletne lub zupełne nazwy domen (ang. *Fully Qualified Domain Names (FQDNs)*) nazywane są te domeny, których nazwy zawierają domenę każdego poziomu, począwszy od lokalnej, aż po domenę główną, czyli root. Łatwo dostrzec analogię do wcześniej wspomnianego systemu pliku w systemach operacyjnych UNIX pomiędzy FQDN a bezpośrednią ścieżką do pliku. Różnicą jest tu jednak sposób odczytywania takiej ścieżki. W systemach DNS najbardziej ogólny węzeł znajduje się na skrajnie prawej pozycji a poruszając się w stronę lewą dochodzimy do kolejnych lokalnych domen.

### 1.2.6 Mapowanie nazw na adres IP

Mapowanie nazw domeny na jej adres IP jest możliwe dzięki plikom strefy DNS, które znajdują się na autorytatywnym serwerze przestrzeni nazw, tzw. *zone files*. Jeden z typów tych plików przechowuje nazwy domen wraz z odpowiadającymi im adresami IP. Gdy klient chce dowiedzieć się pod jaki adres kierować swoje zapytanie, kieruje informacje do serwera autorytatywnego. On odpowiada za znalezienie i przedstawienie mapowania nazwy na adres IP, dokonując przeszukania plików strefy.

### 1.2.7 Mapowanie odwrotne

Baza danych DNS może także zawierać pliki, które umożliwiają mapowania odwrotne, tj. adresu IP na nazwę domeny/hosta. Mechanizm ten może być użyty przy próbie weryfikacji pochodzenia wiadomości. Chcąc wykluczyć próbę oszustwa ze strony prawdziwego nadawcy możemy posiłkować się właśnie odwrotnym mapowaniem. Możemy zweryfikować, czy adres IP mapowania odwrotnego zgadza się z adresem IP z którego nadano wiadomość. Mechanizm ten może być również wykorzystywany do autoryzacji operacji wykonywanych zdalnie.

Odwrotne mapowanie wykorzystuje specjalną domenę *in-addr.arpa*. Z założenia domena ta wykorzystuje adresy IP zamiast adresów domen. Wspomniana domena jest częścią strefy DNS, która umożliwia takie właśnie odwzorowanie. Istotne jest, że adresy w domenie *in-addr.arpa* są zapisywane w specyficzny dla siebie sposób – od najniższego do najbardziej istotnego poziomu. Wynika z tego, że adresy IP w opisywanej domenie są w pewnym sensie zapisywane „od końca”. Powołując się na przykład, założymy, że maszyna ma przypisany adres 10.8.0.32. W plikach strefy dla domeny *in-addr.arpa* adres ten będzie zapisany jako 32.0.8.10.in-addr.arpa. oczywiście z kropką po członie *in-addr.arpa*, która reprezentuje domenę *root*.

### 1.2.8 Wpisy plików strefy

Tak jak wspomniano we wcześniejszych punktach 1.2.6 oraz 1.2.7 działanie całego mechanizmu mapowanej (czy to właściwego czy odwrotnego) może zachodzić dzięki serwerom autorytywnym oraz przechowywanym przez nie plikom strefy DNS.

Z racji, że system DNS dostarcza administratorom wielu funkcji poza standardowym mapowaniem adresów na IP, wpisy w bazach danych mogą być różnych typów. Najpopularniejsze typy to oczywiście te, z którymi można zetknąć się każdego dnia, na przykład wpis typu *A* – tłumaczący nazwę hosta na adres IP w wersji 4, *AAAA* – mapujący nazwę hosta na adres IP w wersji 6, rekordy *CNAME* służące do aliasów, czy *MX* – specyfikujące serwer wymiany elektronicznych dla danej domeny. Oprócz wspomnianych podstawowych typów twórcy standardu dla systemu DNS wprowadzili także mniej znane typy. Opis zbioru najbardziej istotnych wpisów określonych w dokumencie RFC 1035 [67] wraz z ich przeznaczeniem umieszczono w tabeli 1.3.

Rekord	Pełna nazwa	Pełniona funkcja(opis zawartości)
A	Wpis mapowania adresów (ang. <i>Address mapping record</i> )	Określa adres IP wersji 4 dla danego hosta. Służą do konwersji nazwy domen do odpowiednich adresów IP.
CNAME	Nazwa kanoniczna (ang. <i>Canonical Name</i> )	Określa alias. Ruch kierowany do domeny o takiej nazwie przekierowywany jest do domeny wyspecyfikowanej w rekordzie CNAME.

MX	Wymiana wiadomości elektronicznych (ang. <i>Mail exchange</i> )	Na podstawie nazwy domeny znajdującej się w adresie poczty wskazuje serwer poczty elektronicznej, który odpowiedzialny jest za przyjmowanie tych wiadomości. We wpisie znajduje się również priorytet używany wówczas, gdy dostępnych jest więcej serwerów pocztowych. Ponadto, zbiór rekordów w domenie wykorzystywany jest do trasowania wiadomości w protokole SMTP.
NS	Serwer nazw (ang. <i>Name server</i> )	Określa nazwę serwera, który został oddelegowany do obsługi danej domeny.
SOA	Wskaźnik na początek węzłów hierarchii (ang. <i>Start of Authority</i> )	<p>Wpis SOA zawiera informacje dotyczące zarządzania strefą. Jest istotny głównie z punktu widzenia transferu strefy DNS. Składa się z odpowiednich wpisów:</p> <ul style="list-style-type: none"> <li>• primary – adres serwera głównego (ang. <i>primary</i>) dla danej strefy,</li> <li>• adres poczty administratora,</li> <li>• numer seryjny – uaktualniany przy każdej zmianie informacji w pliku strefy,</li> <li>• czas życia – czas w sekundach określający okres odpytywania serwera głównego przez serwer podrzędny o numer seryjny pliku strefy; zapewnia spójność danych na serwerach podrzędnych,</li> <li>• ponowienie żądania – czas, który powinien od czekać serwer podrzędny jeśli serwer główny nie odpowiedział na żądanie (w sekundach); po jego upłynięciu możliwe jest ponowne wysłanie żądania,</li> <li>• czas przedawnienia – czas w sekundach, po którym serwer podrzędny powinien przestać odpytywać serwer podstawowy,</li> <li>• TTL czas przechowywania odpowiedzi typu NXDOMAIN od serwera autorytatywnego (ang. <i>negative caching</i>).</li> </ul>

TXT	Tekst	Wpis przeznaczony do umieszczania dodatkowych informacji dla klienta. Jest to niesformatowany tekst, może być informacją zarówno dla człowieka jak i maszyny.
-----	-------	---

Tabela 1.3: Rodzaje rekordów w bazach danych serwerów przestrzeni nazw. Opis na podstawie [67].

Oczywiście na przestrzeni lat protokoły rozszerzano o kolejne funkcje. Jednym z istotnych wydarzeń było zaproponowanie zmian spisanych w dokumencie RFC 4034 [15]. Opisano tam rozszerzenia protokołu DNS, które podnoszą jego bezpieczeństwo. W tabeli 1.4 przedstawiono najważniejsze z punktu widzenia niniejszej pracy typy rekordów, które zostały wprowadzone w kolejnych dokumentach RFC, których dokładne numery zostały wspomniane przy konkretnych wpisach w tabeli.

Rekord	RFC	Pełna nazwa	Pełniona funkcja(opis zawartości)
AAAA	3596 [85]	Wpis mapowania adresów IPv6	Określa adres IP wersji 6 dla danego hosta. Zasada działania jest taka jak w przypadku rekordu A z jedną różnicą w wersji adresu IP.
DNSKEY	4034 [15]	Klucz publiczny strefy	Przechowuje klucz publiczny strefy. Pozwala klientowi na weryfikację odpowiedzi od serwera, które są podpisane kluczem prywatnym.
RRSIG	4034 [15]	Podpis rekordu (ang. <i>Resource Record Signature</i> )	Przechowuje podpis grupy RRset wygenerowany przy użyciu klucza prywatnego strefy, który odpowiada kluczowi publicznemu. W rozszerzeniu DNSSEC przyjęto założenie, że każdej zabezpieczonej grupie odpowiadał będzie odpowiedni rekord RRSIG. RRSIG nie jest podpisany.
NSEC	3845 [77]	Ang. <i>next secure</i>	W protokole DNSSEC rekord NSEC jest używany do wskazania kolejnego rekordu, który znajduje się w strefie. Między innymi dzięki niemu jest możliwe zapewnienie spójności danych. Wprowadzony głównie ze względu na umożliwienie sprawdzenia autentyczności odpowiedzi negatywnych.
SPF	4408 [93]	Ang. <i>Sender policy framework</i>	Definiuje, które z wiadomości elektronicznych mogą być dostarczane do danej strefy. Odbywa się to poprzez weryfikację adresów poczty. Rekordy te są wykorzystywane do filtrowania spamu oraz wiadomości niebezpiecznych dla odbiorców.

RP	1183 [33]	Osoba odpowiedzialna za domenę	W odróżnieniu od osoby odpowiedzialnej za całą strefę DNS, której adres zawarty jest w rekordzie SOA, wskazuje adres poczty osoby odpowiedzialnej za daną nazwę domenową.
LOC	1876 [26]	Informacje lokalizacyjne	Pozwala na umieszczenie informacji o geograficznym położeniu maszyn bądź sieci.
SRV	2782 [40]	Lokalizacja usług (ang. <i>service location</i> )	Wpis zaprojektowany w celu ułatwienia klientom lokalizowania serwisów, których poszukują w danej domenie. Wpis jest wykorzystywany najczęściej w odniesieniu do protokołów SIP [74] bądź XMPP [75]. Jeśli klient chce sprawdzić, czy na danym serwerze uruchomiona jest usługa protokołu SIP, wystosuje do serwera zapytanie o domenę _sip._tcp.somedomain.com. Nietypowy format zawierający znaki „_” został zastosowany ze względu na uniknięcie kolizji nazw domen wykorzystywanych w rekordach SRV z adresami, pod którymi dostępne są „rzeczywiste” usługi.

Tabela 1.4: Rodzaje rekordów w bazach danych serwerów przestrzeni nazw określonych w rozszerzeniach dokumentu RFC 1035 [67].

## 1.2.9 TLD ARPA

Specyficzną domeną najwyższego rzędu jest domena *ARPA*. Jest to jedna z nielicznych domen ściśle powiązana z dokładnie jedną instytucją. Skrót ARPA wywodzi się historycznie od nazwy jednostki (ang. *Advanced Research Projects Agency (ARPA)*), która brała czynny udział w propagowaniu oraz rozwijaniu sieci Internet. Dzięki tej jednostce powstał między innymi prekursor dzisiejszego Internetu – ARPANET. Aktualnie TLD ARPA jest wykorzystywane jedynie w celach dotykających infrastruktury internetu. Skrót rozwijany jest jako Address and Routing Parameter Area.

Tę domenę wykorzystuje się również w przypadku protokołu RevDNS a konkretniej do tłumaczenia odwrotnego adresów IP na nazwy domen. Każdy z adresów znajduje się w poddomenie .arpa. Są to odpowiednio poddomeny in-addr.arpa dla adresów IPv4 oraz ip6.arpa dla adresów IPv6.

Innym z zastosowań domeny .arpa jest mapowanie numerów telefonów. Sposób integracji sieci Internet wraz z siecią telefoniczną również wykorzystuje zalety systemu DNS oraz pozwala na

identyfikację odpowiednich przestrzeni nazw. Aby odróżnić przestrzeń nazw sieci Internet od sieci telefonicznej wydzielona została domena drugiego poziomu w TLD arpa. Otrzymała ona nazwę e164. Mechanizm mapowania numerów tradycyjnej sieci PSTN na adresy internetowe ma szcze-gólne znaczenie w integracji dwóch technik wykonywania połączenia, a konkretne PSTN oraz VoIP. Mapowanie numerów na adresy internetowe zostało opisane między innymi w RFC3761 [35].

### 1.2.10 Plik strefy DNS

Strefy DNS są przechowywane na serwerach w formie specjalnie sformatowanych plików. To w nich definiuje się oraz umieszcza kolejne wpisy, rekordy, które będą zwracane dla różnego rodzaju zapytań. Plik strefy jest jednym z kluczowych jeśli chodzi o konfigurację serwera DNS. Na listingu 1.1 zaprezentowano przykładowy plik strefy DNS wraz z kilkoma wpisami. Przedstawiony plik jest sformatowany zgodnie z wymaganiami stawianymi przez implementację serwera DNS zawartą w pakiecie BIND [58, 50, 72].

---

```
1 $TTL 2d
2 $ORIGIN example.com.
3 @ IN SOA ns1.example.com. hostmaster.example.←
    com. (
4           2003080800
5           2h
6           15M
7           3W12h
8           2h20M
9           )
10          IN NS ns1.example.com.
11          IN NS ns2.example.com.
12          IN MX 10 mail.example.com.
13 ns1      IN A   192.168.0.3
14 ns2      IN A   192.168.0.4
15 mail     IN A   192.168.0.5
16
17 $ORIGIN us.example.com.
18 @ IN NS ns3.us.example.com.
19          IN NS ns1.example.com.
20 ns3      IN A   10.10.0.24
21 $
```

---

Listing 1.1: Przykładowy plik strefy DNS.

Na listingu 1.1 zaprezentowano definicję domeny *example.com* oraz poddomeny *us.example.com*. Plik rozpoczyna się od definicji czasu aktywności każdego z rekordów (linia 1), który wynosi 2 dni. Kolejnym poleceniem jest wskazanie dla jakiej domeny definiowane są wpisy DNS. Pierwszym wpisem jest rekord SOA. Kolejne pola zawarte w tym rekordzie zostały już opisane w tabeli 1.3. Kolejne wpisy (linie 10 oraz 11) dostarczają informacji o serwerach przestrzeni nazw, które obsługują daną domenę. Następnie umieszczona jest informacja o głównym serwerze pocztowym w domenie. Linie 13–15 zawierają informację o adresach IP przypisanych do usług zdefiniowanych w poprzednich rekordach DNS. Są to odpowiednio dwa serwery przestrzeni nazw (ns1 oraz ns2) oraz jeden serwer poczty elektronicznej (mail).

W linii 17 zapoczątkowana została kolejna logiczna część pliku strefy. Jest to początek definiowania wpisów dla domeny *us.example.com* czyli domeny niższego poziomu wcześniejszej zdefiniowanej *example.com*. Zaprezentowany przypadek jest przykładem delegacji odpowiedzialności za translację nazw do innego serwera DNS. Jest to zrealizowane poprzez umieszczenie wpisu *IN NS ns3.us.example.com*. Informuje on o tym, że serwerem przestrzeni nazw w danej poddomenie jest serwer z prefiksem *ns3*. Dodatkowo umieszczono również informację o serwerze dla domeny wyższego poziomu (*ns1*). Ostatni rekord zaprezentowany na listingu wskazuje adres IP wersji 4. wypunktowanego wcześniej serwera *ns3*.

Zaprezentowana notacja jest tylko jedną z kilku, których można użyć do definiowania plików strefy DNS. W przypadku pakietu BIND istotne jest, że fraza \$ORIGIN <domena> obowiązuje od miejsca, gdzie została użyta aż do kolejnej definicji domeny bądź do końca pliku.

### 1.3 Transfer strefy DNS

Transfer strefy DNS jest typem transakcji DNS, która jest podstawowym elementem mechanizmu odtwarzania bazy danych DNS pomiędzy serwerami systemu. Transfer jako protokołu warstwy 4 używa TCP i przybiera formę komunikacji typu klient-serwer. Zgodnie z założeniem projektantów klient jest podrzędnym lub zapasowym serwerem DNS a serwer, który odpytuje to nadrzędny bądź główny serwer DNS. To, co wysyłane jest w odpowiedzi od serwera głównego jest strefą DNS – część bazy danych przechowywanych na głównym serwerze.

Transfer może odbywać się za pośrednictwem dwóch wyspecyfikowanych rodzajów zapytań – AXFR [66, 55] bądź IXFR [70]. To pierwsze definiuje transfer całej strefy, niezależnie od jej wersji czy zmian które zostały naniesione, zaś drugie transfer „inkrementalny” czyli na podstawie informacji zawartych w otrzymanym zapytaniu klienckim potrafi wydzielić tylko tę część strefy, która się zmieniła i przesłać tylko te rekordy w odpowiedzi.

Niezależnie od wybranego sposobu przesyłania informacji o strefie protokół można określić jako inicjowany przez klienta. Oznacza to, że każda próba pobrania informacji będzie rozpoczęła się od klienta, który wystosuje do serwera odpowiednie zapytanie. Można wyobrazić sobie sytuację, kiedy klient potrzebuje pobrania najbardziej aktualnej informacji o strefie DNS, na przykład jego baza danych jest pusta bądź „wygasła” ważność rekordu SOA określona przez poprzedni

transfer strefy. Oczywiście z uwagi na inicjację komunikacji przez klienta, możliwe jest pobieranie bazy danych DNS cyklicznie czy w interwałach narzuconych przez administratora serwera podległego. Określone zostały także mechanizmy, które pozwalają powiadomić klientów o zaistnieniu zmian w bazie danych na serwerze podstawowym [55, 89], jednak wciąż odpowiedzialność za rozpoczęcie transferu spoczywa na kliencie.

### 1.3.1 Transfer AXFR

Pierwszym wymienionym sposobem dokonania transferu danych jest wystosowanie do serwera zapytania typu 252. – AXFR [55] używając do tego celu między innymi połączenia TCP. Serwer odpowiada na zapytanie kolejnymi wiadomościami, które niosą informację o rekordach zapisanych w strefie DNS. Istotnym faktem jest, że w tym przypadku serwer przesyła wszystkie rekordy jakie są zawarte w podgrupie określonej strefy DNS. Transfer zawsze zaczyna się od rekordu typu SOA, gdzie widnieje między innymi wersja przechowywanej strefy oraz inne informacje, które opisane zostały szerzej w tabeli 1.3. Następnie przesyłane są wszystkie informacje przechowywane na serwerze podstawowym, zaś całość kończy ponownie rekord SOA co sygnalizuje, że cały transfer został zakończony pomyślnie.

Zawarty w rekordzie SOA numer seryjny pozwala na stwierdzenie, czy kopia strefy, która przechowywana jest na maszynie klienckiej wymaga aktualizacji. Dzięki temu możliwe jest ustalenie, czy warto w ogóle wysłać żądanie przesłania części bazy DNS. Pozwala to w wielu przypadkach oszczędzić zbędnej wymiany informacji między serwerami.

### 1.3.2 Transfer IXFR

Drugą możliwością pobrania informacji na temat strefy DNS jest tak zwany transfer inkrementalny IXFR, którego wartość pola QTYPE wynosi 251. Od zapytania AXFR różni się tym, że przesyłane informacje są tylko różnicą pomiędzy kolejnymi wersjami przechowywanymi na serwerze podstawowym. Wymaga to podania w wystosowanym zapytaniu wersji pliku, który przechowywany jest aktualnie na komputerze-kliencie. W odpowiedzi uzyskiwana jest lista zmian, które zostały wprowadzone w kolejnych wersjach danej strefy DNS. Zawarte w niej są rodzaje informacji. Po pierwsze są to rekordy dodane do strefy DNS w danej wersji bazy danych. Po drugie te, które zostały usunięte.

Istotną informacją w kontekście transferu inkrementalnego jest fakt, że odpowiedią na zapytanie IXFR może być odpowiedź AXFR, a przekierowanie takie nosi nazwę *AXFR fallback*. W implementacji serwera BIND [50] możliwe jest zablokowanie takiego przekierowania. AXFR fallback następuje najczęściej w sytuacji, kiedy serwer nie wie jak odpowiedzieć na otrzymane zapytanie IXFR. Może to następować po otrzymaniu wersji strefy wyższej niż ta przechowywana na serwerze podstawowym, albo w momencie, kiedy na serwerze podległym nie ma dostępnej żadnej bazy danych [81].

## 1.4 Podpisy TSIG

Istotnym typem rekordu jest również typ 250. – rekord TSIG (ang. *Secret Key Transaction Authentication for DNS/Transaction Signature*) zdefiniowany w dokumencie RFC 2845 [88]. Używany jest przede wszystkim w protokole DNS aby zapewnić, że informacja przesyłana pomiędzy obiema komunikującymi się stronami faktycznie pochodzi od nadawcy oraz że nie była modyfikowana w trakcie komunikacji. Mechanizm używany jest przede wszystkim do dynamicznych aktualizacji baz DNS oraz do transferów stref pomiędzy serwerem głównym a podrzędnym. Aby komunikacja była kryptograficznie bezpieczna, w protokole wykorzystywane są klucze tajne oraz bezkolizyjne funkcje skrótu.

### 1.4.1 Opis działania TSIG

Rekord TSIG pozwala na wykorzystywanie mechanizmów znanych z protokołu DNSSEC [49] w protokole DNS zaproponowanych w RFC 1035 [67]. Praktyka taka została zaproponowana z powodu częstych problemów z używaniem protokołu DNSSEC [13, 14]. Wprowadzenie korzystania z rekordów TSIG pozwala między innymi na:

1. kontrolowanie aktualizacji stref DNS,
2. zabezpieczenie transferu strefy DNS,
3. zabezpieczenie komunikacji pomiędzy aktorami (na przykład pomiędzy serwerami przestrzeni nazw).

Zgodnie z nazwą, rekord TSIG jest w głównej mierze kontenerem mającym za zadanie przechowywanie podpisu danej wiadomości DNS. Dany podpis, a więc i zawartość rekordu TSIG jest zgodny jedynie z wiadomością dla której go wygenerowano, więc nie ma powodu, dla którego rekordy tego typu powinny być przechowywane. TSIG znajduje się w części dodatkowej APDU DNS. Po odebraniu pakietu zawierającego TSIG, odpowiedni rekord z sekcji dodatkowej zostaje usunięty i zapisany w oddzielnym miejscu w pamięci. Nagłówek wiadomości jest odpowiednio modyfikowany, tak aby pola długości jak i liczby odpowiedzi od serwerów były zgodne z faktycznym stanem po dokonanej modyfikacji. Następnie liczony jest skrót wiadomości z wstępnie przeprocesowanego pakietu i porównywany z podpisem zapisanym uprzednio w innym miejscu w pamięci. W sytuacji, gdy wartość obliczona przez funkcję skrótu jest różna od wartości odebranej w rekordzie TSIG razem z całą odpowiedzią DNS pakiet taki należy odrzucić oraz powiadomić o tym fakcie nadawcę wiadomości. Rekord TSIG niesie także informacje o dwóch czasach: pierwszy – kiedy utworzono skrót oraz drugi – jak długo skrót zachowuje swoją ważność. Weryfikacja odebranego pakietu obejmuje nie tylko wartość funkcji skrótu, ale także opisany czas. Jeśli czas odebrania wiadomości nie zawiera się w okresie jej ważności odsyłany jest komunikat o błędzie. Dodanie czasu utworzenia pakietu było konieczne, aby zabezpieczyć się przed atakami typu powtórzeniowego (ang. *reply attacks*), gdzie atakujący wykorzystuje informacje z podsłuchanego pakiet

ponownie. Wykorzystywanie stempli czasowych wymaga użycia odpowiedniego zegara. Nie jest to problem jeśli maszyna jest podłączona do internetu, ponieważ może być wykorzystany wtedy protokół NTP (ang. *Network Time Protocol*) [65].

Generowanie podpisanej odpowiedzi może mieć miejsce tylko po odebraniu podpisanej zapytania od klienta. Serwer nie może wysłać odpowiedzi zawierającej rekord TSIG jeśli otrzymał niepodpisane zapytanie. Generacja skrótu znajdującego się w odpowiedzi składa się zarówno ze skrótu, który przysłał klient jak i zawartości rekordu TSIG [49].

Użycie mechanizmu podpisywania wiadomości eliminuje problem nieuprawnionego transferu danych. Z pewnością wyklucza użycie algorytmu wykorzystywanego w niniejszej pracy magisterskiej, gdyż łamanie nawet prostych kluczy wielokrotnie zwiększa czas procesowania pojedynczej domeny [49]. Oczywiście zabezpieczenie kluczy używanych do podpisywania wiadomości a także ich długość i jakość są bardzo ważne z punktu widzenia bezpieczeństwa protokołu. Zaleca się, aby długość generowanego skrótu była mniejsza bądź równa długości klucza użytego do podpisania wiadomości.

#### 1.4.2 Wady TSIG

Problemem związanym z wykorzystaniem rekordów TSIG jest przede wszystkim dystrybucja kluczy. Ponadto w systemie DNS rzadko zdarza się, że tylko jeden klient będzie korzystał z interfejsu serwera, więc każdy z klientów powinien posiadać swój klucz, co ponownie prowadzi do problemu dystrybucji, przechowywania i zarządzania [49].

Inną istotną wadą protokołu jest rodzaj zaproponowanej funkcji skrótu tj. HMAC-MD5. Algorytm ten nie jest uważany w dzisiejszych czasach za bezpieczny. Ataki na HMAC-MD5 zaprezentowano między innymi w pracach [38, 92].

Kolejnym problemem a raczej niedoskonałością zaproponowanego rozwiązania jest zupełnie płaska struktura. Oznacza to tyle, że na poziomie podpisanych wiadomości przy użyciu TSIG nie mamy dostępnych żadnych poziomów hierarchii. Jeśli wiadomość spełnia wymogi formalne, czyli ma poprawne stemple czasowe oraz dobrze wyznaczoną wartość funkcji skrótu to traktowana jest dokładnie tak samo jak inna poprawna wiadomość, bez rozróżnienia z jakiego źródła pochodzi. Gdy mówimy o protokole DNS, gdzie jednym z kluczowych elementów jest hierarchiczna struktura musimy liczyć się z faktem, że mechanizm TSIG może być nieodpowiedni.

### 1.5 Inne metody podpisywania wiadomości

Oprócz przedstawionego w podpunkcie 1.4 mechanizmu TSIG postało kilka innych propozycji podpisywania wiadomości w protokole DNS. Propozycje te opracowywane były przede wszystkim dlatego, że TSIG charakteryzuje się pewnymi uciążliwymi wadami opisanymi w podpunkcie 1.4.2 – nie rozwiązuje problemu dystrybucji kluczy, nie uwzględnia poziomów w hierarchii systemu DNS czy wykorzystuje przestarzałą funkcję skrótu HMAC-MD5.

Niektóre propozycje skupiały się jedynie na zabezpieczaniu dynamicznych aktualizacji wpisów rekordów DNS [7], a inne rozwiązywały tylko problemy bezpiecznej, skutecznej i automatycznej dystrybucji kluczy pomiędzy stronami protokołu – serwerem oraz resolwerem [8]. Wymiana kluczy opisana w dokumencie [8] miałaby odbywać się dzięki nowemu typowi rekordu DNS – TKEY (ang. *transaction key*).

Powstawały także rozwiązania rozszerzające TSIG jak na przykład algorytm zaproponowany w RFC3645 [53]. Wykorzystuje się w nim GSS [57](ang. *Generic Security Service*) aby zapewnić bezpieczną u automatyczną dystrybucję kluczy do klientów protokołu TSIG. Bardziej generyczna metoda wymiany kluczy była opisana w RFC2930 [8], gdzie wykorzystanie GSS-API było tylko jedną z możliwości rozwiązania postawionego problemu.

Innym kierunkiem rozwoju protokołu TSIG jest wykorzystywanie innych algorytmów implementujących funkcje skrótu. Spowodowane jest to faktem, że RFC2845 [88] definiuje wykorzystanie jedynie funkcji HMAC-MD5. Jedną z takich prób opisano w dokumencie RFC4635 [9] gdzie zaproponowano zastąpienie przestarzałej funkcji MD5 funkcjami z rodziny SHA. Opisano odpowiednio użycie SHA1 [30] lub algorytmów zaproponowane w projekcie FIPS PUB 180-2, znane później jako SHA-2 [29].

## 1.6 Parkowanie domen

Parkowanie domen to zachowanie, które można określić jako wypożyczenie kompletnej nazwy domeny (FQDN 1.2.5). Odnosi się ono do przypisywania nieużywanych, kompletnych nazw domen do systemów, które mogą zapewnić obsłużenie ruchu generowanego do danej domeny. Takie rozwiązanie znalazło szerokie zastosowanie na przykład w branży marketingowej. Podmioty posiadające domeny mogą je „zaparkować” poprzez specjalne serwisy. Cały proces jest ukierunkowany na pozyskiwanie korzyści finansowych. Użytkownikowi, który wpisuje nazwę zaparkowanej domeny jest wyświetlana specjalna treść, na przykład reklamy, a przychody z marketingu są dzielone pomiędzy firmę oferującą usługę parkowania domeny oraz jej właściciela. Zaparkowane domeny są najczęściej ciągami znaków, które powstały w wyniku błędów wpisywania nazw popularnych domen, bądź wariacjami nazw dotyczących usług finansowych czy bankowych. Rynek parkowania domen okazał się niezwykle dochodowy, czego potwierdzeniem może być raport jednej z firm oferującej takie usługi – Sedo Holding [45].

Sposób, w jaki realizowana jest usługa parkowania domen może sprzyjać również różnego rodzaju nadużyciom. Jednym z przykładów jest tak zwany click-spam. Termin związany jest z głównym wskaźnikiem popularności reklam internetowych – liczbą kliknięć. Click-spam określa nadużycie, gdzie jedna ze stron jest źródłem całego ruchu kierowanego do serwisu reklamy, jednak dzięki pewnym mechanizmom, z punktu widzenia firmy marketingowej jest ona reprezentowana jako wiele rozdzielnych podmiotów. Wykorzystanie parkowanych domen w odniesieniu do click-spamu zostało omówione między innymi w publikacji [24]. Z click-spamem łączy się pośrednio inne zagrożenie, które może nieść za sobą parkowanie domen – wielokrotne przekierowywania po-

między domenami. Przekierowania mogą w pewnym momencie mylić, czy niepokoić użytkownika powodując wrażenie, że nie ma on kontroli nad tym, co przetwarzane jest na jego komputerze. Problemy związane z parkowaniem domen zostały szerzej omówione w dokumencie [82].



## 2. Powiązane prace

Pomimo faktu, że problem „wycieku” danych z serwerów DNS poruszany był już w 2002 roku [12] oraz w późniejszych publikacjach [20], na blogach bądź forach dotyczących testowania infrastruktury pod kątem bezpieczeństwa [2, 94], to nie udało dotrzeć się do żadnych globalnych badań na temat bezpieczeństwa transferu strefy DNS. To właśnie skala badania przeprowadzonego w ramach niniejszej pracy magisterskiej jest główną różnicą w stosunku do wykonanych już analiz. Ponadto, bardzo ważnym założeniem przedstawionym w tej pracy jest chęć nakreślenia skali zjawiska nieuprawnionego transferu danych DNS. Określenie jak duży jest to problem jest konieczne do wydania ewentualnych zaleceń oraz przyporządkowania priorytetu bądź rangi tej podatności. W poniższym rozdziale przedstawiono istniejące, powiązane prace oraz ich różnice w stosunku do opisywanych badań zawartych w niniejszej pracy dyplomowej.

### 2.1 Internet-Wide Scan Data Repository

*Internet-Wide Scan Data Repository* jest to publiczne archiwum danych dostępne głównie pod adresem scans.io [83]. Gromadzone są tam dane badawcze zebrane podczas aktywnych skanów przeprowadzonych w internecie. Repozytorium prowadzone jest przez badaczy z Uniwersytetu z Michigan [28]. Cel pobierania danych nie jest wyraźnie określony przez naukowców, to co zostanie z nich wywnioskowane pozostawia się osobom trzecim. Z punktu widzenia niniejszej pracy, *Internet-Wide Scan Data Repository* może posłużyć jako punkt odniesienia na przestrzeni skanowania AXFR. Na stronie możemy znaleźć wyniki aktywnego skanowania domen z listy Alexa Top 1 Milion [5] pod kątem podatności na nieuprawniony transfer strefy DNS.

Jednym z udogodnień ze strony pracowników Uniwersytetu z Michigan jest udostępnienie wygodnego interfejsu. Zarządcy opisywanego repozytorium danych udostępniają również aplikację, wyszukiwarkę informacji na temat domen przeskanowanych pod różnym kątem w ich badaniach. Cały projekt nosi tę samą nazwę co zespół – Censys [28] i pozwala wyszukiwać informacje zarówno po adresach URL/IP jak i po zawartości odpowiedzi uzyskanych ze skanów.

Różnicą, która jest pomiędzy pracami opisanymi w tym punkcie a niniejszą pracą magisterką jest oczywiście zasięg prowadzonych badań. Ograniczenie się jedynie do miliona najpopularniejszych domen, tak jak to zrobił zespół Censys [28] jest bardzo dużym uproszczeniem. Ponadto, naturalne jest, że najpopularniejsze strony będą przykładały dużo uwagi do odpowiedniego zabez-

pieczenia swoich zasobów, dlatego można się spodziewać, że wyników takiego skanowania będzie mniej niż na równej ilościowo, losowo wybranej próbie.

## 2.2 Projekty open source

Podczas prac nad niniejszą pracą magisterską natknęto się na dwa hobbystyczne projekty dotyczące transferu strefy DNS. Pierwszym z nich jest projekt nazwany AXFR dostępny w serwisie GitHub [43]. W ramach niego zaimplementowane zostało narzędzie umożliwiające równoległe przetwarzanie żądań transferu strefy AXFR. Całość zrealizowano w języku Python, zaś danymi wejściowymi była lista miliona najpopularniejszych domen Alexa Top 1 Milion [5]. Dodatkowo, w ramach projektu przeprowadzono analizę uzyskanych wyników. Niestety wyniki nie zostały opisane, rezultaty uzyskane podczas prowadzenia tego projektu zostały jedynie pogrupowane w odpowiednich plikach. Z nazw plików można wnioskować, czego dotyczyła dana analiza. Wnioskując w ten sposób można założyć, że w ramach projektu sprawdzano jak wiele wpisów w pozyskanych strefach odnosi się do: rekordów A/AAAA, rekordów SPF oraz SPF all, usług takich jak git, svn oraz Jenkins. W odróżnieniu od opisanego projektu, w niniejszej pracy magisterskiej przeprowadzono badania na dużo większą skalę. Dodatkowo możliwe będzie porównanie wyników uzyskanych w obu skanowaniach. Ponadto, oczekiwany wynikiem jest bardzo duża dysproporcja w ilości uzyskanych odpowiedzi. Wynikać powinna ona nie tylko z powodu dużo szerszego skanowania. Można się spodziewać, że najczęściej odwiedzane domeny, z racji na swoją popularność oraz generowane zyski, nie pozwalają na transfer strefy DNS osobom trzecim.

Drugim projektem jest również implementacja znaleziona w serwisie GitHub o roboczej nazwie Python-AXFR-Test [69]. W tym projekcie autor skupił się na dostarczeniu implementacji skanera domen ukierunkowanego na poszukiwanie domen umożliwiających transfer strefy DNS. Został zaimplementowany również prosty mechanizm wielowątkowości i właśnie ta cecha jest największą korzyścią względem programu dig [51], który oferuje bardzo podobną funkcjonalność do zaprezentowanego programu. Niestety, autor udostępnił jedynie program umożliwiający testowanie domen. Nie zostały opublikowane żadne wyniki czy dyskusja tego, co udało się przeskanować. Projekt miał istotny wpływ na początkowe etapy tej pracy magisterskiej. Możliwe było wykorzystanie części zaprezentowanej implementacji w jednym ze skanerów zrealizowanych podczas prac. Niemniej jednak, zakres niniejszej pracy magisterskiej jest dużo szerszy od zakresu prac nad Python-AXFR-Test, a program ostatecznie uzyskany podczas realizacji pracy magisterskiej zapewnia lepszą wydajność oraz więcej funkcji.

Oba projekty opisane w tym punkcie są blisko powiązane z ideą zaprezentowaną w tej pracy dyplomowej. Niewątpliwie istotny wpływ na analizę zebranych danych miały informacje udostępnione przez autora pierwszego projektu. Implementacja dostępna w repozytorium projektu Python-AXFR-Test [69] w dużym stopniu przyczyniła się do osiągnięcia końcowej wersji skanera. Mimo wszystko, zasięg badań hobbystycznych zaprezentowanych w tym rozdziale jest dużo mniejszy, niż obszar zrealizowanych badań w ramach pracy dyplomowej.

## 2.3 DNS Response Policy Zone

DNS Response Policy Zone jest metodą zarządzania odpowiedziami na konkretne zapytania DNS. Funkcja ta została wprowadzona w wersji 9.10 pakietu BIND oraz od tamtego wydania funkcjonalność dostępna jest w każdym kolejnym. DNS RPZ jest często nazywane Firewalliem DNS. Nazwa ta jest dobrym określeniem działania całego systemu. Dzięki niemu, można definiować reguły odpowiedzi, na przykład na podstawie źródła żądania bądź informacji, które ono zawiera. Administratorzy podejmują się definiowania reguł odpowiedzi między innymi ze względu na obawy dotyczące udostępniania informacji niezaufanym maszynom. Innymi powodami, dla których zarządcy chcą definiować reguły odpowiedzi protokołu DNS to na przykład próba ograniczenia ruchu (na przykład ruchu wychodzącego) pewnym grupom użytkowników. Innym przykładem zastosowania DNS RPZ jest ograniczenie dostarczania informacji o wystawionej usłudze tylko zaufanej grupie użytkowników. Przykłady te można mnożyć, wiele przypadków znanych z klasycznych usług fire-walli ma uzasadnienie także dla firewalla DNS. Dokładne wymagania odnoszące się do zarządzania regułami odpowiedzi na zapytania DNS spisano oraz udostępniono w dokumencie [90]. Publikacja ta potwierdza, że istnieje potrzeba posiadania usługi czy mechanizmu, dzięki któremu możliwa będzie dokładniejsza kontrola przepływu informacji w protokole DNS. DNS RPZ to tylko jedno z możliwych rozwiązań, dzięki któremu możliwa jest ochrona danych zawartych w strefach DNS. W odróżnieniu od prezentowanej pracy magisterskiej, temat niebezpieczeństw wynikających z po-wszechnego udostępniania informacji strefowych nie został tam szeroko zaprezentowany, a jedynie wspomniany jako jeden z powodów propozycji danego rozwiązania.

## 2.4 DNS as a service

*DNS as service* to pojęcie wykorzystywane do opisywania usług świadczonych w obrębie protokołu DNS. Dosłownym tłumaczeniem tego terminu jest *DNS jako usługa*. W rozumieniu protokołu DNS jest to wystawienie interfejsów klientowi, tak, aby mógł on dodać interesujące go rekordy.

*DNS as a service* spotykane jest najczęściej w kontekście unikania ataków DDoS jednak mimo to można przyjrzeć się temu zjawisku również przy okazji analizy bezpieczeństwa transferu strefy DNS. Określenie *Distributed Denial of Service* jest w tym przypadku swego rodzaju chwytem marketingowym, ponieważ najbardziej efektywnie działa na wyobraźnię klientów. Niemniej jednak, bezpieczeństwo systemów DNS to nie tylko unikanie ataków DDoS.

Tym, co faktycznie sprzedają firmy oferujące usługi *DNS as a service* jest przede wszystkim infrastruktura, ale także pewnego rodzaju bezpieczeństwo informacji oraz pewność dostępności systemu DNS. Dobrym przykładem jest w tym przypadku oferta firmy Nexusguard [56]. Oferuje ona obsługę oraz zabezpieczenie systemu DNS swojego klienta. Wprowadzenie danych na temat strefy DNS klienta może odbywać się na dwa sposoby:

1. poprzez panel klienta,

2. poprzez transfer strefy DNS z serwera klienta.

W kontekście tej pracy magisterskiej pierwszy przypadek nie jest nadzwyczaj interesujący. Klient uzupełnia dane o rekordach DNS swojej domeny czy też strefy i cała odpowiedzialność za rozwiązywanie kolejnych zapytań jest przeniesiona na maszyny działające pod nadzorem firmy oferującej usługę. Dużo bardziej interesującym przypadkiem jest wymieniony jako drugi transfer strefy z serwera klienta. Wydzielona część infrastruktury firmy jest wtedy traktowana jako zapasowy serwer DNS(*ang. slave*) podczas gdy serwerem podstawowym (*ang. primary*) jest maszyna klienta. W taki sytuacji transfer strefy DNS jest dużo bardziej bezpieczny. Wynika to z wymagań narzuconych przez organizację, mowa między innymi o mechanizmie TSIG (*ang. Transaction Signature*) opisanym w rozdziale 1.4.1 oraz wykorzystaniu połączenia szyfrowanego opartego na SSL pomiędzy serwerem podstawowym a podrzędnym. Jeśli chodzi o bezpieczeństwo dostępu do rekordów DNS poprzez panel kliencki, to możliwe jest wykorzystanie dwuetapowej weryfikacji użytkownika.

Warto dodać, że *DNS as a service* jest również w ofercie kilku innych, na przykład Akamai [84], CloudFlare [22] czy Imperva [47]. Obecność usług takich jak *DNS as service* sugeruje, że jest to bardzo istotne, aby odpowiednio zabezpieczać nie tylko główne maszyny i usługi w swojej sieci ale i niemal każdy element, który działa w naszym systemie. Dodatkowo sugeruje to, że często administratorzy czy klienci wolą przenieść odpowiedzialność za utrzymywanie takiego systemu na firmę trzecią w zamian za odpowiednie ich wynagrodzenie.

## 2.5 DNS Enumeration

Rozszerzenie DNSSEC (*DNS Security Extentions*) zostało zaprezentowane w roku 1997 [10] jako odpowiedź na poważne luki w bezpieczeństwie systemów DNS opisane w dokumencie [63]. W dużym uproszczeniu rozszerzenie to pozwala na zweryfikowanie, czy odpowiedź na zapytanie wysłane do serwera pochodzi na pewno od niego a nie od atakującego system. Wyspecyfikowanie standardu DNSSEC było problematyczne, a organizacje pracujące nad nim wielokrotnie go zmieniały. Podyktowane to było różnego rodzaju ograniczeniami technicznymi. Jednym z takich ograniczeń technicznych jest zjawisko *DNS Enumeration* (lub *Zone Enumeration*). Jest to podatność w sieci polegająca na wypisaniu wszystkich urządzeń, które do tej sieci należą. W standardzie DNSSEC umożliwiono *zone enumeration*, którego wynikiem były nawet urządzenia sieciowe (na przykład rutery) znajdujące się w danej strefie. Zjawisko nie jest luką krytyczną, jednak może prowadzić do bardziej złożonych ataków. Cały proces ataku na daną strefę zabezpiezioną rozszerzeniem DNSSEC określone jest jako *zone walking*.

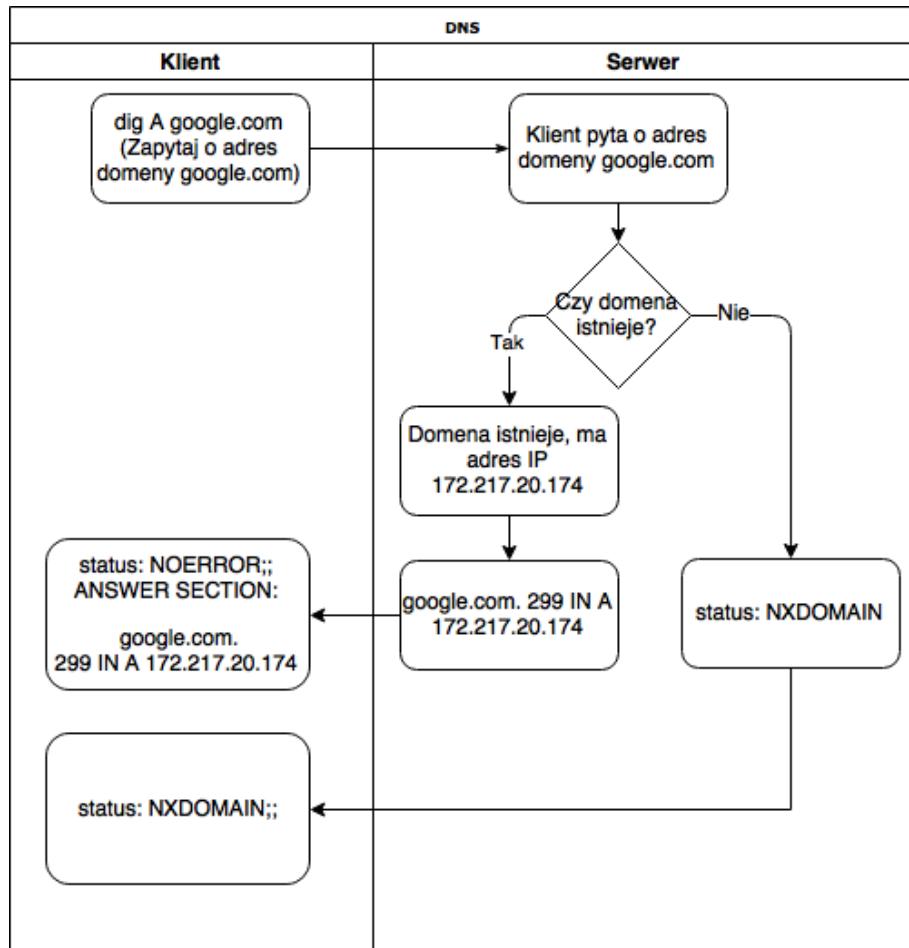
Problem listowania strefy DNS wynika z różnego traktowania odpowiedzi pozytywnej oraz negatywnej od serwera. Odpowiedź pozytywna jest wysyłana w przypadku, gdy dana domena istnieje oraz przypisano jej adres IP. Przykładowo:

1. Klient: Jaki jest adres domeny www.przyklad.pl?

2. Serwer: Domena www.przyklad.pl ma adres 1.2.3.4.

Jeśli chodzi o odpowiedź negatywną, to przykład opisano poniżej:

1. Klient: Jaki jest adres dla domeny nieistnieje.przyklad.pl?
2. Serwer: Domena nieistnieje.przyklad.pl nie istnieje.



Rysunek 2.1: Przepływ informacji podczas odpytywania serwera przestrzeni nazw o domenę.

Zarówno przypadek odpytywania o domenę istniejącą, jak i przypadek przeciwny zostały pokazane na schemacie 2.1. Wyróżniono na nim stronę serwera oraz klienta oraz zaprezentowano części wiadomości, które są istotne dla konkretnego przypadku

Oba przytoczone powyżej przypadki są inaczej traktowane w protokole DNSSEC. Dla przykładu, gdy odpowiedzią jest adres IP istniejącej domeny, serwer autorytywny przechowuje określony, skończony zbiór podpisanych rekordów. Podpisy są tworzone przy użyciu klucza prywatnego

danej domeny. Istotny jest fakt, że podpisy rekordów nie są obliczane w czasie rzeczywistym, a jedynie przechowywane razem z innymi rekordami w bazie danych DNS. Zaletą takiego rozwiązania jest oczywiście redukcja obliczeń wykonywanych przez serwer autorytatywny a poza tym, tylko jeden z serwerów musi być w posiadaniu klucza prywatnego, więc dystrybucja klucza jest w dużym stopniu uproszczona. Dodatkowo w przedstawionym wcześniej modelu nie ma potrzeby, aby weryfikować tożsamość każdego z serwerów w systemie DNS, co ponownie upraszcza komunikację, zmniejsza obciążenie zasobów w sieci oraz ilość wymienianych informacji.

Problem listowania strefy DNS pojawia się wraz z charakterystyczną, negatywną odpowiedzią (w specyfikacji określana jako NXDOMAIN). Nie jest możliwe, aby odpowiadać na pytanie o niepoprawną domenę wcześniej przygotowaną wiadomością z obliczonym skrótem, ponieważ może to prowadzić do skutecznego ataku powtórzeniowego (ang. *reply attack*). Nie jest możliwe obliczenie wszystkich skrótów, dla każdej z poddomen, bo takich przypadków jest zbyt wiele. Ciężko także zrezygnować z niewątpliwej zalety przytoczonej w poprzednim akapicie – braku konieczności obliczania skrótów w czasie rzeczywistym. Pomyśłodawcy rozwiązania opisanego w RFC4034 [15] zaproponowali algorytm, dzięki któremu możliwe będzie utrzymanie substytutu podpisanej wiadomości NXDOMAIN. Cały system zachowuje również zalety przedstawione w poprzednim akapicie. Rozwiązanie zakłada, że rekordy w strefie są uporządkowane, każda para rekordów jest podpisana i tworzy rekord NSEC, a każdy rekord podpisany kluczem prywatnym. Jeśli serwer odbierze zapytanie o domenę, która nie istnieje, odpowiada rekordem NSEC pary domen, które w uporządkowanej liście znajdują się odpowiednio przed i po odpytywanej domenie oraz związane z nimi podpisy. Rozwiązanie zapewnia, że tylko serwer autorytatywny powinien być obdarzony zaufaniem oraz umożliwia wstępna generację podpisów. Niestety wprowadza dodatkowo efekt uboczny, w postaci umożliwienia *zone enumeration*, czyli wylistowania strefy. Przykładowa odpowiedź na zapytanie o nieistniejącą poddomenę została zaprezentowana na listingu 2.1.

---

```
1 $ kdig +dnssec +multiline ddadasds.example.com
2 ;; ->>HEADER<<- opcode: QUERY; status: NXDOMAIN; id: 22793
3 ;; Flags: qr aa rd; QUERY: 1; ANSWER: 0; AUTHORITY: 8; ADDITIONAL: 1
4 ;; QUESTION SECTION:
5 ;; ddadasds.example.com. IN A
6 ;; AUTHORITY SECTION:
7 example.com. 3600 IN SOA dns1.example.com.
8 example.com. 3600 IN RRSIG SOA 13 2 3600 20170128184611
9      ( 5134 example.com. nqiEgM+kVBDeBI== )
10 ;; Matching record for hash of example.com;
11 0sc7qshrek878fcmnag1.example.com. 3600 IN NSEC3 1 0 0 AABB
12      ( CPDHD7GK40NGDKRU8CQ8 NS SOA MX RRSIG DNSKEY NSEC3PARAM )
13 0sc7qshrek878fcmnag1.example.com. 3600 IN RRSIG NSEC3 13 3 3600
14      ( 5134 example.com. 2JicIoTH3WkgAjbP/ehmTv== )
15 ;; Covering record for hash of ddadasds.example.com;
```

```

16 jftj44t4kqppke20mukr.example.com. 3600 IN NSEC3 1 0 0 AABB
17     ( MSC7QSHREK878FCM8GD7 A AAAA RRSIG )
18 jftj44t4kqppke20mukr.example.com. 3600 IN RRSIG NSEC3 13 3 3600
19     ( 5134 example.com. VfFQfho5sQ8QVWOqsrxYN6== )
20 ;; Covering record for hash of *.ddadasds.example.com;
21 cpdhd7gk40ngdkru8cq8n.example.com. 3600 IN NSEC3 1 0 0 AABB
22     ( J1VSBFDBU38SMLNPIMM A AAAA RRSIG )
23 cpdhd7gk40ngdkru8cq8n.example.com. 3600 IN RRSIG NSEC3 13 3 3600
24     ( 5134 example.com. lcDsoeVGuq3rvezN2oW74x== )
25 ;; Received 773 B
26 $

```

---

Listing 2.1: Odpowiedź na zapytanie DNSSEC o nieistniejącą domenę (na podstawie [27]).

Zagadnienie listowania stref DNS samo w sobie często było podmiotem dyskusji. Początkowo wydano dokument, że nie jest to błędem, że umożliwia się wypisanie bazy danych DNS [13]. Później jednak wypracowany został kompromis [54], że w pewnych przypadkach znajomość wszystkich domen może powodować dodatkowe niebezpieczeństwa. Przykłady, które przytoczono w RFC5155 [54] to na przykład dobre źródło danych wejściowych, które mogą posłużyć jako prawdopodobne adresy mailowe w kampaniach spamowych bądź jako informacje o infrastrukturze wykorzystywane podczas rekonesansu DNS. Ponadto podatność nazywana *zone enumeration* może wpływać negatywnie na organizacje zajmujące się rejestrami DNS. Często są one zobowiązane do nieujawniania danych przechowywanych w swoich rejestrach. Rekord NSEC umożliwia przeprowadzenie *DNS enumeration* na strefach tych organizacji a następnie wykonanie zapytań whois [23] w celu pozyskania informacji o osobie rejestrującej domenę.



# **3. Implementacja**

Przeanalizowanie problemu postawionego w niniejszej pracy magisterskiej wymagało implementacji własnych narzędzi. Dostępne są w sieci programy umożliwiające przeprowadzenie rekonesansu DNS i są bardzo dobre (dig [51], axfr-tool [69], robotex [59]) jednak nie są to skanery, które umożliwiają globalne skanowanie, a więc nie zapewniają oczekiwanej wydajności dla dużych zbiorów danych. Wynika to z faktu, że są one nastawione głównie na analizę konkretnego przypadku. Temat ten został szerzej opisany w punkcie 3.1. Narzędzia te są używane głównie przez pentesterów, których typowym zadaniem jest przetestowanie pod kątem bezpieczeństwa na przykład pewnej aplikacji, gdzie liczba serwerów DNS jak i testowanych domen jest mocno ograniczona. Proces ten sam w sobie jest niezwykle złożony i często wymaga nawet tygodni pracy, dlatego czas w jakim wykonywał się będzie program sprawdzający podatności DNS nie jest aż tak istotny. Oczywiście mówimy o różnicach w czasie w granicach do kilku minut. W podejściu prezentowanym w tej pracy założono, że przeanalizuje się możliwie jak najwięcej domen dostępnych w sieci internet. Dlatego też różnica procesowania jednej pary domena – adres IP serwera autorytatywnego już nawet na poziomie kilku sekund pozwala zaobserwować bardzo duży skok wydajności.

Dojście do rozwiązania optymalnego nie było trywialne. Początkowe rozwiązania opierały się na próbach rozszerzenia dostępnych narzędzi o managera zarządzającego procesami, w których uruchamiane były zewnętrzne narzędzia – głównie dig. Rozwiążanie takie miało szereg zalet, na przykład było bardzo skalowalne i łatwe do zrealizowania. Niestety okazało się, że menedżer zarządzający procesami wymaga zbyt dużej ilości zasobów, a głównie pamięci RAM, dlatego wymuszone zostało zrezygnowanie z tego podejścia.

## **3.1 Przegląd dostępnych narzędzi**

Społeczność internetu oferuje bardzo szeroką gamę narzędzi umożliwiających pozyskiwanie informacji na podstawie protokołu DNS. Bardzo użytecznym i przydatnym narzędziem jest w tym przypadku program dig. Wspomniany program ma jednak znaczącą wadę – jest wydajny jedynie przy niewielkiej liczbie odpytywanych domen. Dodatkową niedogodnością jest, wbrew pozorom, fakt, że program wchodzi w skład pakietu Open Source bind. Ten, jak każde oprogramowanie na wolnej licencji, cierpi na szereg niedogodności z tym związanych. Najbardziej prozaicznym problemem jest fakt, że oprogramowanie jest tworzone przez wiele osób, więc bardzo trudno zastosować jeden

standard kodowania, gdyż każda z osób ma swój preferowany. Poza tym, dużą wadą jest trudność wprowadzania zmian w tego typu oprogramowaniu, wynikającą zarówno z punktu poprzedniego jak i ze złożoności programu, którą cechuje się dig w tym momencie.

Istnieją także inne pakiety implementujące w dość wydajny sposób klienta systemu DNS jak np. `pjlib` [60]. Również w tym przypadku można borykać się z problemami wynikającymi z założeń przyjętych przez twórców tego oprogramowania. Konkretyzując to stwierdzenie - wspomniana biblioteka z założenia miała być wykorzystywana do protokołu SIP, a więc implementacja klienta DNS weszła w jej skład tylko i wyłącznie dlatego, że twórcy jej potrzebowali jako narzędzi do zrealizowania innych celów. Implikuje to fakt, że pobieranie wiadomości jest niekompletne na płaszczyźnie typów wiadomości. Jednym z nich jest typ nr 252, czyli AXFR, który jest jednym z kluczowych elementów rekonesansu na podstawie protokołu DNS.

Innymi narzędziami, które można wykorzystać przy rekonesansie DNS są skrypt języka Python zrealizowane przy okazji hobbyistycznych projektów. Mowa tu o projektach o nazwach AXFR [43] oraz Python-AXFR-Test [69]. Ich kod źródłowy można znaleźć w serwisie GitHub. Mimo, że prezentowana przez nie wydajność wciąż nie pozwalała na wykorzystanie ich w ostatecznej implementacji serwera, to pomogły one w zrozumieniu jak technicznie wygląda transfer strefy AXFR. Dzięki temu możliwe było szybsze nakreślenie specyfiki implementowanych narzędzi oraz późniejsze wykorzystanie pierwszego poziomu abstrakcji do napisania rozwiązania docelowego.

Projekt robotex [59] jest natomiast rozwiązaniem dostępym przez przeglądarkę stron WWW i nie udało się uzyskać informacji na temat ewentualnego interfejsu, który można byłoby wykorzystać na potrzeby pracy magisterskiej. Narzędzie umożliwia dostęp do wielu informacji na temat domeny, głównie z punktu widzenia protokołu DNS, jednak jedną formą uzyskania ich jest wpisanie adresu domeny, która jest poddawana rekonesansowi. Rozwiązanie jest przez to bardzo dobre dla różnego rodzaju testerów, jednak ma niską wartość jeśli chodzi o skanowanie większego zbioru domen.

Krótkie podsumowanie każdego z narzędzi zawarto w tabeli 3.1. Opisano w niej krótko ich zalety oraz wady, skupiając się głównie na kwestiach wykorzystania tych rozwiązań w przeprowadzonych badaniach.

Narzędzie	Zalety	Wady
dig	<ul style="list-style-type: none"> <li>• łatwo dostępny</li> <li>• przejrzysty dla użytkownika</li> <li>• dobrze przemyślany interfejs jak i prezentacja danych wyjściowych</li> </ul>	<ul style="list-style-type: none"> <li>• brak jednolitej struktury kodu źródłowego</li> <li>• duży, trudny do modyfikacji projekt</li> <li>• zapewnienie wyższej wydajności możliwe niemal tylko poprzez skrypty powłoki</li> <li>• brak możliwości zapisu danych wyjściowych do pliku bezpośrednio przez program</li> </ul>
AXFR oraz Python - AXFR-Test	<ul style="list-style-type: none"> <li>• implementacja w całości w Pythonie</li> <li>• łatwo rozszerzalny</li> <li>• w projekcie AXFR [43] dokonano prostych analiz stref pobranych w wyniku skanowania</li> <li>• obsługa plików – lista domen czytana z plików oraz zapisywanie pobranych danych</li> </ul>	<ul style="list-style-type: none"> <li>• niska wydajność</li> <li>• ograniczenie wielowątkowości wynikające ze specyfiki języka Python</li> </ul>

Robotex	<ul style="list-style-type: none"> <li>pozyskiwanych jest wiele informacji na temat skanowanej domeny</li> <li>ciekawa prezentacja zależności między elementami (domain tree)</li> </ul>	<ul style="list-style-type: none"> <li>ręczne rozpoczęwanie skanowania</li> <li>brak możliwości automatyzacji skanowania</li> <li>przydatny tylko do skanowania kilku domen</li> </ul>
Biblioteka pjlib	<ul style="list-style-type: none"> <li>wydajna implementacja w języku C</li> <li>projekt w dużym stopniu jednolity w kwestii standardów kodowania</li> </ul>	<ul style="list-style-type: none"> <li>specyficzne mechanizmy wykorzystane w projekcie</li> <li>brak obsługi wszystkich przypadków użycia protokołu DNS</li> </ul>

Tabela 3.1: Podsumowanie istniejących narzędzi.

## 3.2 Zaimplementowane narzędzia

Przeprowadzenie globalnego skanu niosło za sobą szereg konsekwencji. Zakładając, że średnio należy przeznaczyć na skanowanie jednej pary (domena, adres IP) około 3 sekund okazało się, że przeprocesowanie całego zbioru danych wejściowych (ok. 5 miliardów par) zajęłoby odpowiednio:

$$3(s) * 5 * 10^9 = 25 * 10^8(min) = \frac{25}{60} * 10^8(h) = 41.67 * 10^6(dni)$$

Opierając się na opisany wyżej rozumowaniu, podjęto decyzję, że należy procesować pary (domena, adres IP) równolegle. Podjętych zostało kilka prób implementacji odpowiedniego narzędzia umożliwiającego przeprowadzenie badań w zadowalającym czasie:

- narzędzie oparte na programie dig skryptach powłoki bash,
- menadżer procesów programu dig oparty na skryptach języka Python,
- skaner AXFR (C-AXFR) wspomagany skryptami powłoki bash.

Szczegółowo zostaną one opisane w kolejnych punktach tego rozdziału.

### **3.2.1 Program dig zarządzany skryptami powłoki**

Pierwszym rozwiązaniem, które powstało w ramach pracy nad opisywanym problemem był skaner AXFR oparty na programie dig [50]. Program skupia się wokół protokołu DNS i umożliwia wykonywanie zapytań różnych typów [58]. Ponadto, jest prosty w obsłudze, daje duże możliwości jeśli chodzi o budowanie zapytań DNS oraz bardzo często dostępny jest domyślnie w wielu dystrybucjach systemów operacyjnych Linux.

Odpowiedzi uzyskane dzięki programowi miały być zapisywane do plików tekstowych. Każda, potencjalnie uzyskana odpowiedź miała być zapisywana do oddzielnego pliku tekstowego, aby wykluczyć równoczesne użycie tego samego zasobu przez wiele procesów. Rozwiązanie tego typu działa oraz z powodzeniem przeskanowano kilka tysięcy domen, jednak okazało się zbyt wolne do zastosowania na szerszą skalę a także występowały kłopoty z zapotrzebowaniem na zasoby (mowa tu głównie o pamięci operacyjnej). Bliźniacze podejście zostało zastosowane w opisywanym wcześniej projekcie skanowania domen z listy *Alexa 1 Milion* [83].

### **3.2.2 Menedżer procesów zarządzany skryptami języka Python**

Kolejną próbą rozwiązania problemu ogromnej liczby danych do przeskanowania była implementacja skanera w języku skryptowym Python. Próba ta podyktowana była kilkoma istotnymi aspektami. Pierwszym z nich jest implementacja tak zwanych „programowych” wątków w standardowej bibliotece tego języka. Umożliwiło to implementację mechanizmu, który pozwolił sprawdzić, czy maszyna dysponuje takimi zasobami, które pozwolą na uruchomienie kolejnego wątku skanującego. Dodatkowym atutem jest tu także możliwość wywoływania poleceń powłoki systemu Linux ze skryptu języka Python, a więc możliwe było wykorzystanie istniejącej już implementacji programu dig [50].

Niestety podobnie jak we wcześniej opisywanym przypadku, problemem okazały się ograniczenia czasowe spowodowane niewystarczającą ilością zasobów. Program został przetestowany podczas kilkudniowego skanowania i zdołano odpowiednio:

- przeanalizować kilka milionów par (domena, adres IP),
- pozyskać dane do wstępnej analizy podatności w skali globalnej,
- dwukrotnie zawiesić maszynę, na której wykonywano badania.

Pomimo, że implementacja w języku Python nie okazała się na tyle wydajna aby wykorzystać ją w badaniach przeprowadzonych w ramach tej pracy magisterskiej, jest to dobrze narzędzie dla pasjonatów bezpieczeństwa sieciowego, którzy chcą skanować wiele serwerów. Podejście takie zaprezentowane jest w kilku projektach, które można znaleźć w serwisie github na przykład Python-AXFR-test [69] lub skanowanie domen z Alexy wraz ze wstępnią analizą wyników [43]. Liczba osób, które śledzą wymienione wyżej projekty znajduje się w przedziale kilkudziesięciu

osób, więc można wnioskować, że nawet małe projekty realizowane z zakresu tej tematyki są w pewnym stopniu interesujące dla społeczności.

Można przypuszczać, że oba projekty zrealizowane w języku Python [69, 43] oferują podobną wydajność do rozwiązań zaimplementowanego podczas realizacji pracy magisterskiej, jednak nie były prowadzone pomiary pod tym kątem. Brak bezpośredniego porównania wynika z faktu, że prace Stephena Haywooda [43] były prowadzone niemal równolegle z realizacją niniejszej pracy magisterskiej, zaś skrypt Python-AXFR-test [69] jest jednowątkowy i nastawiony głównie na test pojedynczych serwerów, więc jego wykorzystanie wymagałoby implementacji bardzo podobnego mechanizmu, który opisywany był na początku tego rozdziału.

### 3.2.3 Skaner C-AXFR

Ostatecznie zdecydowano się na napisanie własnego narzędzia umożliwiającego transfer strefy DNS. Wybór padł na język programowania C [52] w wersji standardu C11 [36]. Oczywiście aplikacje pisane w języku C nie należą do rozwiązań prostych, jednak są bardzo dobrym kompromisem pomiędzy wysokim poziomem abstrakcji modeli programistycznych i łatwością dostępu do interfejsów sieciowych. Ponadto język C charakteryzuje się dość dobrą szybkością działania, ze względu na komplikację kodów źródłowych do kodu binarnego. Kryterium szybkości działania było jednym z kluczowych przy implementacji, co pokazały poprzednie próby implementacji.

Rolę danych wejściowych programu mogą pełnić odpowiednio:

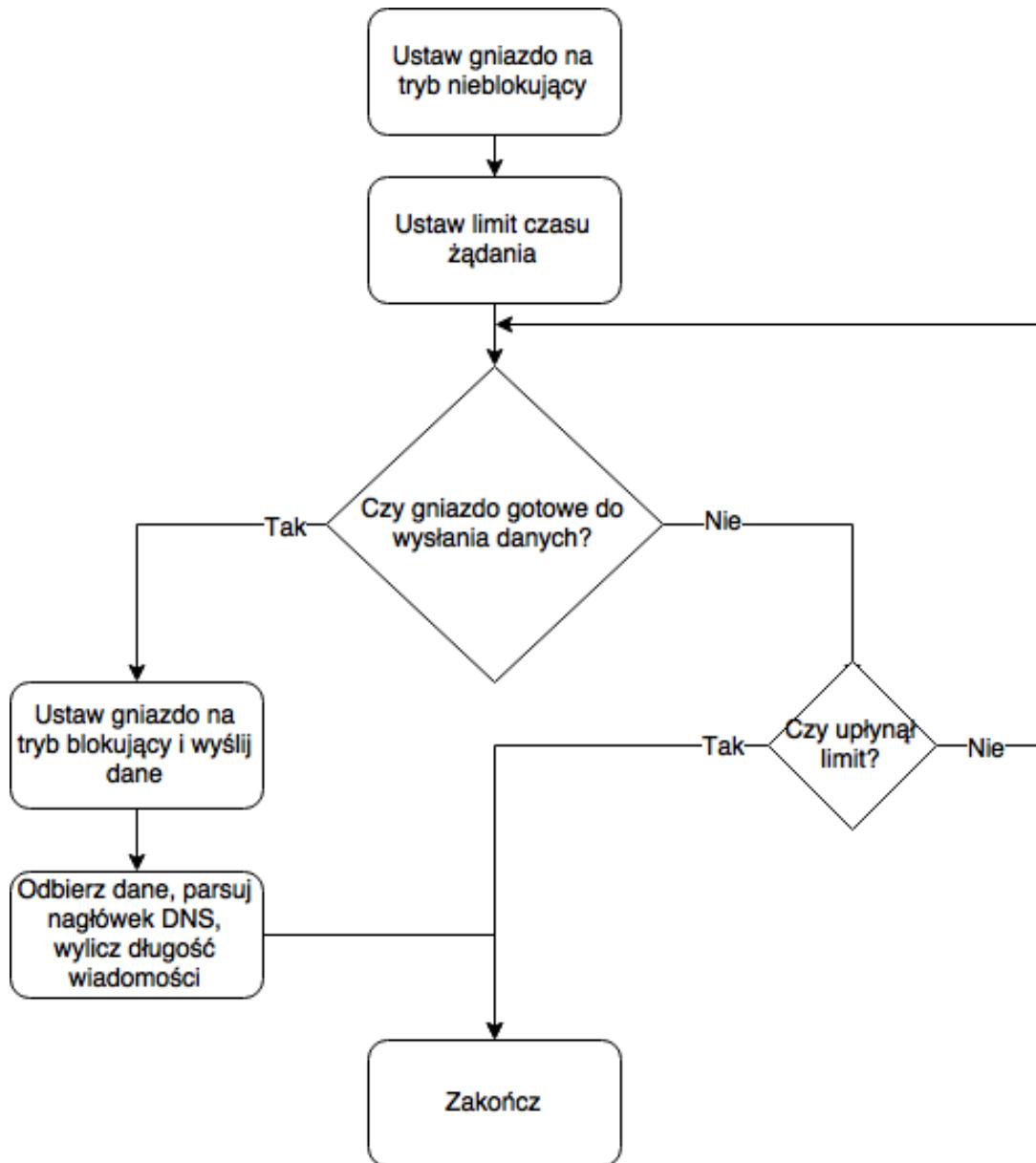
1. plik tekstowy w formacie *adres\_domeny|IP\_serwera\_autorytatywnego*,
2. para parametrów wejściowych podanych w odpowiedniej kolejności - *adres\_domeny* oraz odpowiadający *IP\_serwera\_autorytatywnego*.

Zdecydowano się na takie rozróżnienie z prostego powodu. Najistotniejszą kwestią prezentowanego rozwiązania było zachowanie podobieństwa do programu dig, który jest wykorzystywany przez wiele osób. Dodatkowo, dużo łatwiej można testować logikę programu na małych porcjach danych i opcja numer 2. została w pewnym sensie funkcją debugową. Jako wspomniane dane testowe najczęściej wykorzystywana była domena *zonetransfer.me* [94], której autor specjalnie umożliwia transfer strefy, aby uzmysłowić innym ludziom zagrożenie wynikające z niepowołanego korzystania.

Logika programu jest bardzo prosta. Początkowo budowana jest jednostka APDU protokołu DNS z odpowiednimi danymi, a w szczególności z poprawnie ustawioną flagą *qtype* odpowiedzialną za identyfikację rodzaju zapytania DNS. Co ważne, mechanizm AXFR wymaga używania nietypowego dla DNS protokołu TCP.

Komunikacja z serwerami autorytatywnymi została oparta na blokujących gniazdach TCP. Dodatkowo, zaimplementowany został autorski mechanizm pozwalający na wykorzystywanie własnych limitów czasu żądania przed wysłaniem pakietu TCP. Wymagało to przełączania gniazda na nieblokujący tryb pracy. Następnie wykorzystano funkcję *select()*, która zwraca informację o tym,

czy możliwe jest rozpoczęcie pisania/czytania danych z instancji gniazda TCP. Dokładny algorytm działania został przedstawiony na rysunku 3.1.



Rysunek 3.1: Algorytm dynamicznego ustawiania limitu czasu żądania dla gniazda TCP w blokującym trybie pracy.

Program umożliwia parsowanie najbardziej popularnych i najczęściej używanych rekordów RR. Zostały one spisane w tabeli 3.2 wraz z identyfikatorami, które reprezentują je w protokole DNS.

Lp.	Typ rekordu	ID rekordu
1	A	1
2	AAAA	28
3	CNAME	5
4	HINFO	13
5	TXT	16
6	NS	2
7	SOA	6
8	PTR	12
9	MX	15
10	RP	17
11	AFSDB	18
12	LOC	29
13	SRV	33
14	NAPTR	35
15	RRSIG	46
16	NSEC	47
17	DNSKEY	48

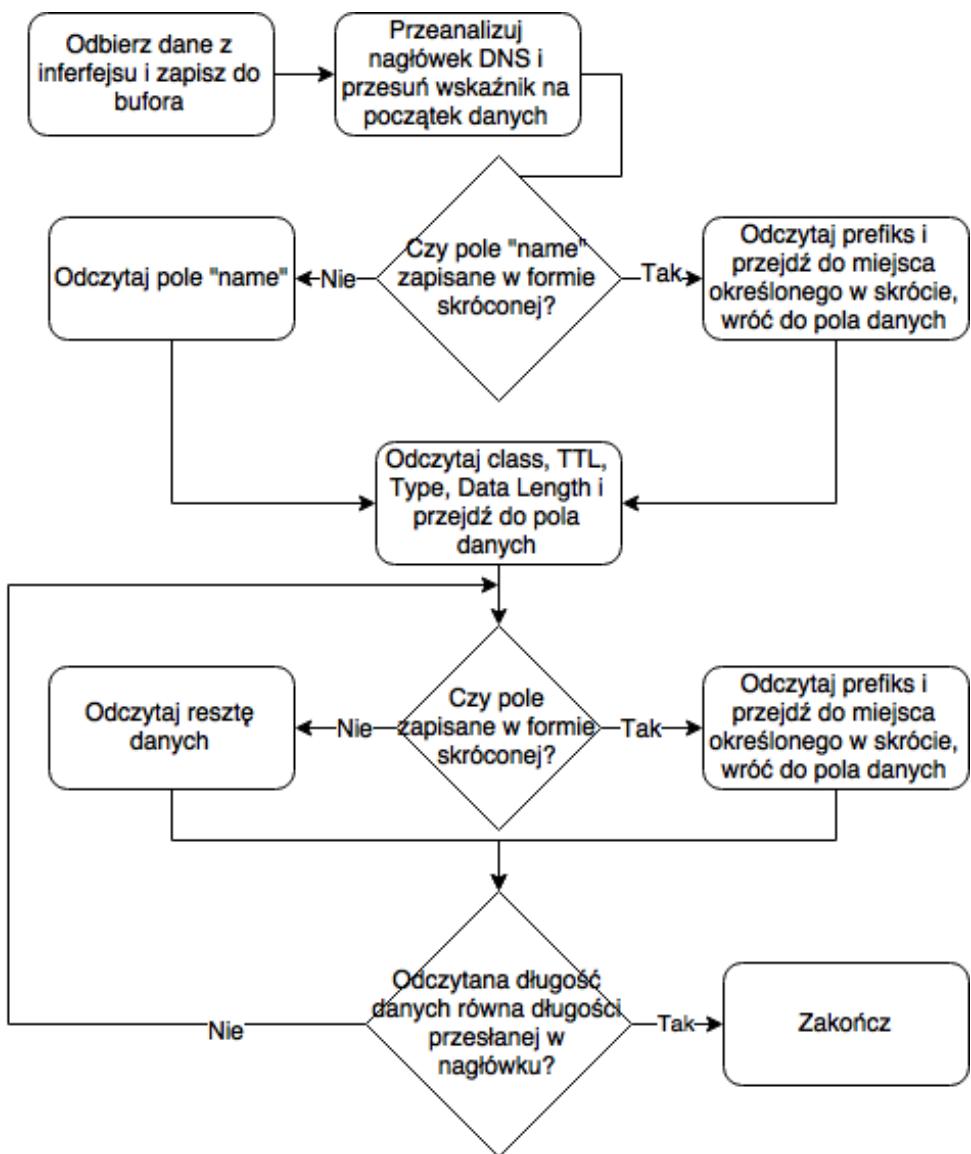
Tabela 3.2: Rekordy DNS obsługiwane w skanerze wraz z identyfikatorami.

Algorytm postępowania w przypadku odebrania pakietu danych przedstawiony jest na rysunku 3.2. Dane odbierane z interfejsu sieciowego są zapisane w systemie szesnastkowym, zgodnie z zaleceniami znanyymi dokumentu RFC 1035 [67].

Dodatkowo, jeśli zostanie odebrany typ, który nie został przewidziany w implementacji parsera to wszystkie dane odczytane z gniazda zostaną przeniesione w formie zapisu heksadecymalnego do pliku z odpowiedzią. Zapewnia ta kompatybilność z innymi rozszerzeniami protokołu DNS oraz zabezpiecza przed błędym wykluczeniem niektórych informacji ze skanowania. Jeśli pojawi się nieznany wcześniej typ rekordu z bazy danych DNS, to będzie można interpretować go podczas analizy konkretnych odpowiedzi od serwera.

Tak jak wcześniej wspomniano, transfer strefy DNS wykorzystuje protokół TCP, który nie jest tak powszechnie stosowany jeśli chodzi o protokół Doman Name System. Wpływ to pośrednio na to, jakie wyniki możemy uzyskać przy próbie odpytania serwera o jego strefę. Mowa tu o trzech charakterystycznych sytuacjach, mianowicie:

1. brak możliwości nawiązania połączenia na warstwie 4 (TCP) na porcie 53,
2. uzyskanie połączenia w rozumieniu protokołu TCP na porcie 53 i brak możliwości transferu danych,



Rysunek 3.2: Algorytm postępowania w przypadku odebrania APDU protokołu DNS.

### 3. uzyskanie połączenia TCP na porcie 53 i pomyślny transfer danych.

Początkowo założono, że uzyskanie już samego połączenia TCP z serwerem może być ciekawym przedmiotem badania. Interfejs serwera autorytywnego danej domeny powinien umożliwiać swoim klientom tylko kilka podstawowych operacji, jak na przykład translację nazwy domenowej komputera ze swojej strefy na jego adres. Do takich działań połączenie TCP nie powinno być potrzebne. Dodatkowo, port 53 na którym działa DNS jest zarezerwowany tylko dla tego protokołu, więc nie powinno udostępniać się na nim innych usług. Dowodzi to temu, że sytuacja, gdy możliwe jest nawiązanie połączenia TCP na porcie 53 jest nienaturalna i niezgodna z panującymi dobrymi praktykami. Właśnie dlatego, w pierwszej implementacji skanera założono, że program

będzie tworzył puste pliki w przypadku opisanym w tym akapicie. Niestety, zjawisko to okazało się tak powszechnie, że powstały duże problemy związane z wydajnością skanera. Mowa tu o dostępie poszczególnych procesów do dysku twardego maszyny na której były one uruchomione. Dlatego też zdecydowano się na zapisywanie informacji jedynie o domenach, które odpowiadają na zapytanie AXFR a sytuację umożliwiania nawiązywania sesji TCP na porcie 53 pozostawia się do analizy podczas przyszłych badań.

Ostatecznie przyjęto rozwiązanie, które ignoruje fakt nawiązywania połączenia TCP oraz odpowiedzi od serwerów, w których nie ma żadnych rekordów. Pozwoliło to na niemal dwukrotny wzrost wydajności zaimplementowanego skanera.

### 3.3 Dane wejściowe

Dane wejściowe programu zostały pozyskane dzięki współpracy z TU Delft [34] w formie par (domena, adres IP serwera autorytarnego). Dane, które składały się na plik wejściowy zostały pobrane z kilku źródeł:

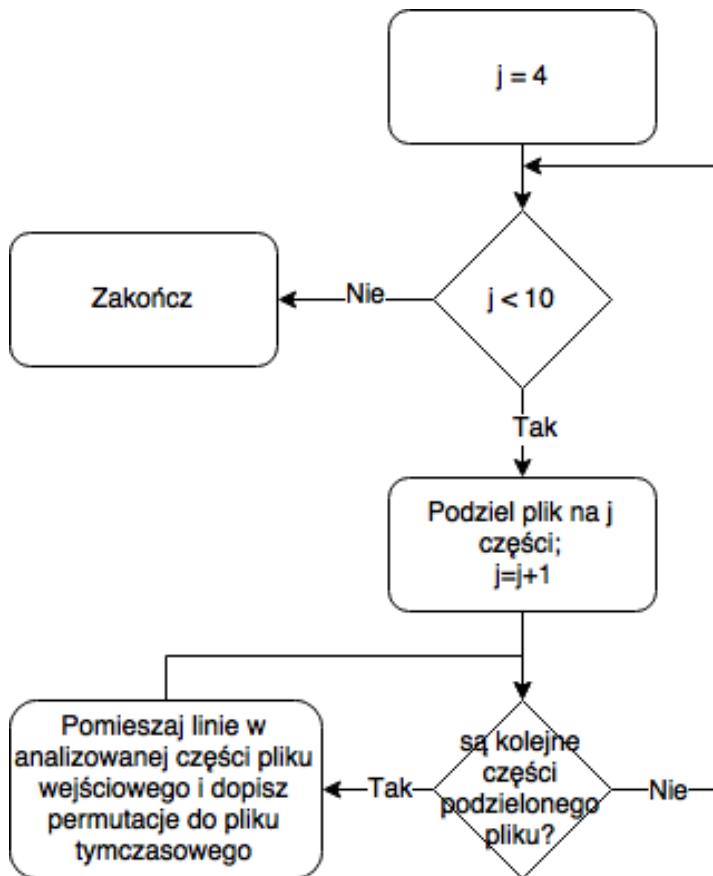
- DNS-DB [78],
- zapytań ANY dostępnych przez serwis scans.io [19],
- plików strefy z jednego dnia dla TLD: *org, info, se, net, US, org, com, name*, wszystkich domen *gTLD*,
- listy Alexa 1M [5] z grudnia 2016.

Następnym krokiem przygotowującym dane wejściowe było wygenerowanie par *domena adres\_NS\_IP*. W kolejnym kroku odfiltrowano adresy arpa, adresy IP sieci prywatnych. Odfiltrowane zostały również adresy oraz domeny dostawców, którzy zgłosili się z prośbą do zespołu z TU Delft, aby ich sieci nie były skanowane podczas badań [61]. Gromadzenie danych rozpoczęło się w styczniu 2015 roku, a zakończyło w grudniu 2016 roku.

Lista w takim formacie przechowywana była w pliku tekstowym. Zgromadzono w nim, zgodnie z tym co zostało przyjęte w obliczeniach 3.2, ponad 5 miliardów wpisów uporządkowanych w kolejności alfabetycznej, a ich łączny rozmiar przekraczał 130 gigabajtów. Powoduje to oczywiście szereg problemów związanych z procesowaniem takich plików.

Jedną z niedogodności, które wiążą się z rozmiarem danych wejściowych był problem z wyborem losowej próby do kolejnych procesów skanera. Pożądane jest, aby plik wejściowy do każdego z procesów skanujących nie był w żaden sposób uporządkowany, a najlepiej gdyby dane w nim zawarte były wybrane losowo. Nie można założyć, że wszystkie procesy skanowania zakończą się bez żadnych błędów. Jeśli więc dane wejściowe byłyby uporządkowane a proces zakończył się błędem wprowadzona zostanie dodatkowa, błędna korelacja między brakiem odpowiedzi od serwera a nazwą domeny.

W systemach Linux dostępny jest domyślnie program *shuf* [32], który dokonuje permutacji na konkretnych liniach w pliku, jednak niemożliwe było wykorzystanie go wraz z plikiem z danymi wejściowymi z powodów ograniczeń wydajnościowych. Program nie był w stanie zaalokować odpowiedniej liczby komórek w pamięci komputera. Aby zasymulować element losowości zastosowano algorytm zaprezentowany na diagramie na rysunku 3.3 oparty na dzieleniu całego zbioru danych wejściowych oraz permutowaniu kolejnych linii w podgrupach o różnej wielkości.



Rysunek 3.3: Algorytm mieszanego linii w dużym pliku wejściowym.

Następnie, plik został podzielony na mniejsze, po 25000 linii w każdym. Liczba linii w pojedynczym pliku została ustalona arbitralnie, tak aby widoczny był postęp w skanowaniu. Przeprocesowanie 25 tysięcy linii odpowiada maksymalnie około 28 godzinom przy założeniu, że dla każdej domeny upłynie limit czasu żądania.

## 3.4 Środowisko uruchomieniowe

Każda z aplikacji była rozwijana w podobny sposób. Cały proces można podzielić na 3 główne etapy:

1. rozwijanie oprogramowania – dodawanie nowych funkcji do skanera oraz proste testy funkcjonalne,
2. uruchomienie w warunkach testowych – sprawdzanie poprawności działania systemu w dłuższej perspektywie (do kilkunastu godzin),
3. uruchomienie w środowisku docelowym – właściwe skanowanie maksymalizujące wydajność.

Ze względu na wygodę, każdy z etapów był uruchamiany na innej maszynie. Dostępność sprzętu była ograniczona, dlatego nie udało się zapewnić, że każda z faz zostanie uruchomiona na takim samym urządzeniu. Faza pierwsza realizowana była na komputerze z procesorem dwurdzeniowym Intel Core i3-4150, 8GB pamięci RAM oraz 1TB dyskiem twardym w zestawieniu ze 120GB dyskiem SSD. Etap uruchomienia systemu w środowisku testowym przebiegał na maszynie z procesorem Intel Core2Duo E6420 z dwoma rdzeniami, 4GB pamięci RAM oraz 320GB dyskiem twardym. Etap ostatni, właściwe skanowanie domen odbywało się na serwerze dedykowanym wypożyczonym od firmy OVH. Zaopatrzony był on w czterordzeniowy procesor Intel Xeon E3-1230v6, 16GB pamięci RAM oraz 2x4TB dysk twardym.

Każda z maszyn działała pod kontrolą systemu operacyjnego opartego na dystrybucji systemu Debian. Rozwijanie oprogramowania prowadzono na systemie operacyjnym Linux Mint w wersji 17.2. Środowisko testowe działało pod kontrolą systemu Debian 9 „Stretch” zaś docelowo wykorzystano dystrybucję Ubuntu 16.04. Każdy z wykorzystanych systemów operacyjnych był jego 64 bitową wersją. Podejście do problemu w zaprezentowany sposób pozwoliło na szereg udoskonalień. Pierwszy etap jest dość oczywisty – nie ma sensu uruchamiać całego systemu na serwerze, jeśli nie mamy pewności, że poszczególne komponenty w samym skanerze nie są odpowiednio zaimplementowane. Mylące może być wprowadzenie etapu uruchomienia systemu w środowisku testowym. Powodem przyjęcia takiego rozwiązania były kwestie finansowe. Nie mając pewności, że system działa dobrze w długiej perspektywie, nie warto było wypożyczać dedykowanego serwera jedynie do testowania. W momencie, gdy można było domniemywać, że system działa stabilnie przez dłuższy okres, można było wypożyczyć serwer i skupić się na uzyskaniu jak największej wydajności skanera.

## 3.5 Metodyka badań

Skaner został zaprojektowany w takie sposób, aby potrafił przeskanować kilka domen podczas jednego uruchomienia. Założenie wynika z ograniczeń wydajnościowych, które zauważono przy próbie implementacji skanera, którą opisano w punkcie 3.2.1.

Skanera nie zaprojektowano tak, aby wspierał wielowątkowość. Może się okazać, że implementacja uruchamiania wielu wątków już w programie będzie jeszcze bardziej wydajna niż proponowana w niniejszej pracy. Mimo to, opisywana propozycja zakłada realizację wielowątkowości

poprzez uruchomienie wielu procesów programu w systemie linux, każdy z inną listą domen do przeskanowania jako argumentem wejściowym. Generacja poszczególnych list została opisana we wcześniejszym podpunkcie 3.3.

Pełen scenariusz skanowania zaprezentowano poniżej.

1. Pobierz listę do skanowania.
2. Wygeneruj N podzbiorów rozłącznych dla listy.
3. Dla każdego podzbioru uruchom proces skanera z odpowiadającą mu listą.

W prezentowanym przypadku liczba podzbiorów była bardzo duża, więc zdecydowano się na ograniczenie liczby uruchamianych procesów skanera jednocześnie. Eksperymentalnie ustalono, że liczba instancji skanera na poziomie 15000 jest akceptowalna zarówno ze względu na wydajność jak i stabilność. Ustalone ograniczenie zostało zrealizowane dzięki skryptom powłoki bash – cyklicznie sprawdzano ile uruchomionych jest aktualnie instancji skanera, jeśli liczba była mniejsza od 15000, to uruchamiano procesy w liczbie takiej, aby suma uruchomionych procesów skanera osiągała zdefiniowaną górną granicę. Aby uniknąć konfliktów zapisu do plików ustalono, że każdy z uruchomionych procesów będzie zapisywać odebrane informacje do oddzielnych katalogów. Katalogi były indeksowane, ich nazwa była postaci „iter\_numer\_iteracji”. Umożliwiło to wyeliminowanie dzielenia zasobów pomiędzy procesami, co przełożyło się na wydajność całego rozwiązania. Pogrupowanie danych wyjściowych w katalogach umożliwiło też prostsze analizowanie zebranych informacji. Możliwa była analiza wielu folderów jednocześnie, co pozytywnie przełożyło się na czas wykonywania się poszczególnych operacji.

## 3.6 System powiadomień

W ramach badań przeprowadzonych podczas realizacji niniejszej pracy magisterskiej został zaimplementowany i uruchomiony system powiadamiania administratorów domen. System bazuje na informacjach pobranych podczas skanowania podatności serwerów na zapytania typu AXFR. W rekordzie SOA przesyłanym na początku i na końcu wiadomości przesyłany jest adres osoby odpowiedzialnej za daną domenę. Jego miejscem jest pozycja numer 2. w rekordzie SOA z tą różnicą, że pierwszy znak „.” napotkany w adresie jest zamieniany na znak „@”. W wyniku tej zamiany uzyskiwany jest adres mailowy, przypisany do administratora domeny.

System powiadomień zaimplementowano w systemie operacyjnym z rodziny systemów UNIX. Opiera się on głównie na skryptach powłoki *bash*. Algorytm wyodrębnia rekordy SOA i wypisuje je do pliku. Następnie plik sortowany jest ze względu na adres mailowy administratora domeny, po czym grupuje się te domeny, za które odpowiada ten sam adres pocztowy. Kolejnym krokiem jest pogrupowanie domen oraz utworzenie odpowiedniego pliku tekstowego. Zawarte są w nim domeny oraz adresy serwerów które umożliwiają transfer strefy dla danej domeny. Nazwa pliku odpowiada adresowi mailowemu zarządcy domeny.

Po otrzymaniu zbioru plików opisanych powyżej uruchamiany jest proces, którego zadaniem jest wysłanie wiadomości pod określone adresy. Skrypt, który automatyzuje opisaną czynność napisano w języku Python [37]. Wysyłane wiadomości podpisywane są kluczem PGP [18], aby odbiorca mógł zweryfikować, że wiadomość została odebrana dokładnie w takiej formie jak ją przygotowano i wysłano.

## **4. Badania eksperymentalne i uzyskane wyniki**

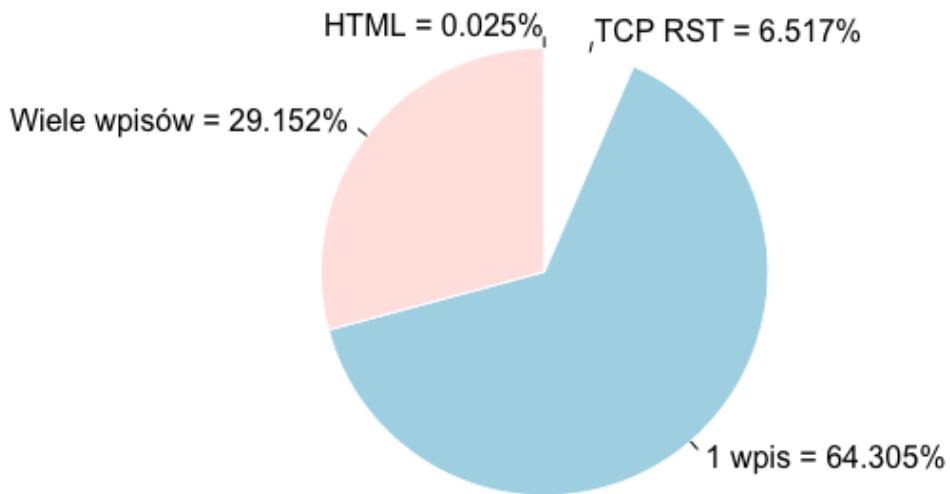
Przeprowadzenie skanowania domen z dostarczonego zbioru danych zajęło 11 dni, co poskutkowało zebraniem około 30GB informacji na temat analizowanych serwerów. Przeskanowano cały plik wejściowy liczący 5032117237 pozycji. Skanowanie przeprowadzone było w dniach od 21. kwietnia 2017 roku do 1. maja 2017 roku włącznie.

Obciążenie skanera zwiększano stopniowo w ciągu kolejnych dni przeprowadzania skanowania. W ten sposób można było uniknąć przeciążenia skanera oraz uzyskać jednocześnie zadowalającą wydajność. Okazało się, że możliwe jest uruchomienie większej liczby procesów niż przewidywano na etapie projektowania. Przełożyło się to na krótszy czas skanowania.

### **4.1 Typy odebranych odpowiedzi**

Wśród przeanalizowanych par domen i serwerów można zaobserwować kilka charakterystycznych typów odpowiedzi. Odpowiedzi mogą być dzielone na kategorie ze względu na różne kryteria. Pierwszym kryterium, które będzie brane pod uwagę w niniejszej pracy jest rozmiar pliku przechowującego informacje pobrane z serwera autorytatywnego. Jest to motywowane faktem, że w istocie im większy jest plik z odpowiedzią, tym spodziewamy się, że zawiera on więcej informacji na temat odpytywanej domeny. Wykres 4.1 przedstawia względna liczebność każdego z typów odebranych wiadomości.

Pierwszym typem jest odpowiedź, która zajmowała na dysku specyficzną ilość miejsca – 25123 bajty. Zaobserwowano, że uzyskano 626288 odpowiedzi tego typu. Rozmiar pliku jest swego rodzaju skutkiem ubocznym uproszczonej implementacji skanera. Założono, że zawsze zostanie odebrany pakiet DNS. Podyktowane było to ograniczonym czasem w którym implementowano skaner oraz faktem, że problem ten można w łatwy sposób obejść poprzez odfiltrowanie odpowiednich plików w katalogu z wynikami. Trudno było przewidzieć wszystkie specyficzne przypadki towarzyszące transferowi strefy DNS. Zaobserwowana sytuacja jest właśnie jedną z tych nietypowych sytuacji i nawet program dig nie odwzorowuje idealnie zachowania jakie powinno nastąpić w takaiej sytuacji. Przyczyną utworzenia opisanego wcześniej pliku nie jest jednoznacznie określona. Zostały podjęte próby ustalenia czym spowodowane jest takie zachowanie. W takich samych przy-



Rysunek 4.1: Wykres liczebności poszczególnych zbiorów typów odpowiedzi.

padkach program dig zwraca jedynie rekord DNS SOA i komunikat o błędzie (ang. *communications error: end of file*). Zachowanie programu dig w dużym stopniu przypomina przekierowanie zapytania IXFR na AXFR (ang. *AXFR fallback*) opisane między innymi w RFC1995 [70]. Wywołanie mechanizmu AXFR fallback następuje w sytuacji, kiedy numer wersji pliku strefy przysłany do serwera jest wyższy niż numer wersji aktualnie na nim przechowywany. Na różnego rodzaju forach [3] czy w serwisach internetowych [4], problem który został opisany pojawia się najczęściej jako problem implementacji oprogramowania PowerDNS [17]. Jednak nie ustalono dokładnie jaki jest powód wysłania pakietu powodującego takie zachowanie w odpowiedzi na zwykłe zapytanie AXFR. Wiadomo, że przyczyną powstania pliku tego typu było odebranie od serwera autorytywnego pakietu TCP RST (ang. *TCP reset packet*) co szerzej opisano w dalszej części tego rozdziału.

Kolejnymi specyficznymi grupami jeśli chodzi o rozmiar pliku z odpowiedzią są już typowe odpowiedzi na zapytania AXFR. Najmniejszy rozmiar mają oczywiście odpowiedzi, które zawierają jedynie rekord SOA i jest to dopuszczalna odpowiedź na zapytanie AXFR. Następnie, wraz ze zwiększającą się liczbą wpisów w pliku strefy DNS, zwiększa się rozmiar odpowiedzi. Nie przekłada się to jednak bezpośrednio na informacje, które można wyodrębnić z takich plików strefy. Zdarza się bowiem sytuacja, w której znaczną część pliku strefy DNS zajmują wpisy podpisów cyfrowych RRSIG, których długość przekłada się na rozmiar plików. Dodatkowo, podpisy genero-

wane mogą być dla każdego rekordu oddzielnie (co szerzej opisano w podpunkcie 1.4), dlatego też wpływa to bardzo znacząco na rozmiar otrzymywanej odpowiedzi.

Ostatnią grupą są odpowiedzi wyraźnie niestandardowe w kontekście protokołu DNS – pakiety TCP, które zawierają w sobie tekst formatowany zgodnie z językiem znaczników HTML. Odpowiedzi te w żadnym stopniu nie wydają się powiązane z protokołem DNS, zawierają informację o błędzie protokołu HTTP. Protokół ten nie powinien być uruchamiany na porcie 53, który został domyślnie przypisany do protokołu DNS.

## 4.2 Odebrane adresy IPv4 oraz IPv6

Częstym argumentem, który pojawia się podczas dyskusji na temat bezpieczeństwa transferów plików bazy danych DNS jest ujawnianie adresów IP. W niniejszym podrozdziale skupiono się głównie na adresach IP w wersji 4. ponieważ jest to wciąż dominujący typ adresacji w Internecie [39]. Okazuje się, że pomimo obaw o wycieki adresów przez dokonywanie nieuprawnionego transferu nikt nie zbadał dokładnie jak duża może być skala tego zjawiska. Umożliwiły to badania przeprowadzone w tej pracy magisterskiej.

Aby nie ograniczać się tylko do jednego typu adresacji urządzeń, zostały oddzielone i przeanalizowane dwa zbiory danych – dla IPv4 oraz dla IPv6. Głównym celem analizy adresów IP było ustalenie systemów autonomicznych(*ang. AS – Autonomous System*), z których pochodzą określone maszyny. Następnie możliwe było, na podstawie numerów AS, określenie kraju w którym dane maszyny są zlokalizowane.

### 4.2.1 Analiza AS

Podstawową informacją, którą uzyskać można wykorzystując pobrane dane z transferów AXFR, jest przynależność otrzymanych adresów IP do konkretnych grup autonomicznych. Grupy te określają przynależność do sieci, które zarządzane są przed tego samego operatora sieciowego, a wykorzystuje się je głównie w protokołach routingu. Informacje związane z systemami autonomicznymi można, w głównej mierze, znaleźć w RFC1930 [42]. Zbiór danych został podzielony ze względu na wersję protokołu IP używanej przy adresacji konkretnych maszyn. Przeprowadzono analizę systemów autonomicznych w rozdzielnych grupach, dla wersji 4 protokołu IP oraz dla wersji 6.

Jak już wcześniej wspomniano (rozdział 4.2), wciąż dominującym sposobem adresacji urządzeń w sieci jest adresacja IP w wersji 4 [39]. Do podobnych wniosków można dojść analizując zbiór danych zebranych podczas skanowania. Okazuje się bowiem, że znaczna większość adresów IP, które zostały pobrane w wyniku skanowania należy do zbioru adresów IP w wersji 4. Dane liczbowe prezentują się w następujący sposób:

1. znaleziono 1695653 adresów publicznych adresów IPv4 – oznacza to, że odfiltrowano te adresy, które zostały określone jako adresy sieci prywatnych w RFC1918 [73],

2. znaleziono 127424 adresów IPv6, które są różne od adresu „localhost” (::ffff:1) – zgodnie z wymaganiami postawionymi w RFC4291 [44].

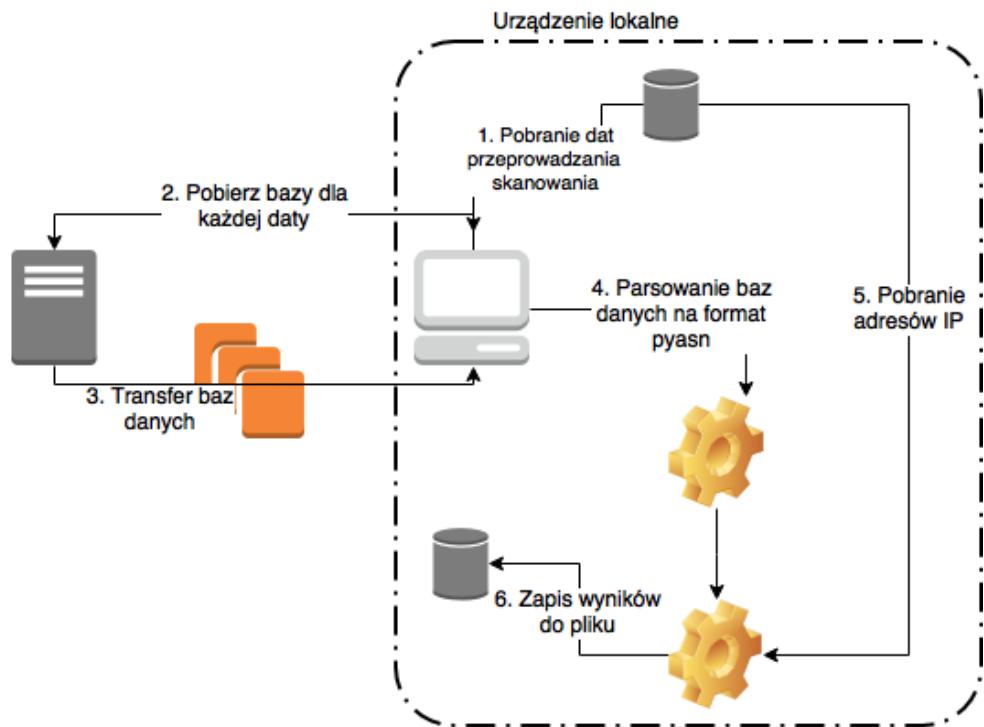
#### 4.2.2 Sposób określenia AS

W celu określenia numerów systemów autonomicznych adresów IP pozyskanych podczas skanowania zaimplementowane zostało kolejne narzędzie umożliwiające takie mapowanie. Społeczność udostępnia kilka sposobów dopasowania adresów IP do odpowiadających im systemów autonomicznych. Podczas badań zapoznano się z dwoma z nich, które opisano poniżej:

1. API udostępnione przez Team Cymru [6],
2. biblioteka języka Python *pyasn* [31].

Pierwszy ze sposobów jest prostszy w użyciu, gdyż wystarczy użyć narzędzia netcat [1], gdzie argumentami jest odpowiednia strona internetowa Team Cymru [6] oraz ścieżka do pliku, w którym znajdują się w kolejnych liniach adresy IP. Każdy z plików rozpoczyna się słowem kluczowym „begin” oraz kończy słowem kluczowym „end”. Sposób ten jest wygodny, jednak może budzić pewne wątpliwości. Najbardziej istotną wydaje się być kwestia dotycząca głównie adresacji IP w wersji 4. – dynamiczne przydzielanie adresów IP różnym operatorom [46]. Problem związany jest pośrednio z kończącą się pulą adresów IPv4. Prowadzi to do zmian przynależności adresów IPv4 do systemów autonomicznych. Dynamikę tych zmian można obserwować między innymi pod adresem [86].

Wątpliwości, które wynikły z wykorzystania udostępnionego API były bezpośrednią przyczyną szukania innego, optymalnego rozwiązania. Takim okazało się wykorzystanie biblioteki *pyasn* [31]. Umożliwia ona zaimplementowanie narzędzia, które uwzględnia datę, gdy dany adres IP został zescanowany. Mapowanie adresu IP na AS w konkretnym dniu jest możliwe dzięki wykorzystaniu wielu baz danych. Okazuje się, że w praktyce, dla każdej z dat udostępniony jest oddzielny plik z bazą danych, dzięki której możliwe jest tłumaczenie adresów IP na ASy. Możliwe jest pobranie bazy danych, która najlepiej odpowiada zebranym pomiarom. W przypadku niniejszej pracy magisterskiej pobierane są pliki dla każdego dnia, w którym przeprowadzone zostało skanowanie. Pliki są dystrybuowane wraz z biblioteką. Użytkownik musi przygotować pliku tekstowego z datami, dla których powinna zostać pobrana baza danych. Następnie bazy są przetwarzane na pliki w formacie odpowiednim dla biblioteki *pyasn*. Aby zautomatyzować opisany proces, zaimplementowano odpowiednie narzędzie w języku skryptowym bash, którego działanie zaprezentowano na diagramie 4.2. Pobiera ono daty z odpowiednich katalogów z wynikami skanów, następnie tworzy odpowiedni plik tekstowy, który później wykorzystywany jest jako plik wejściowy dla biblioteki *pyasn*. Na jego podstawie kopowane są bazy danych dla konkretnych dni. Pobrane bazy danych są następnie konwertowane do takiego formatu, którego wymaga narzędzie. W kolejnym kroku dokonuje się właściwego odwzorowania adresów IP na odpowiadające im ASy.

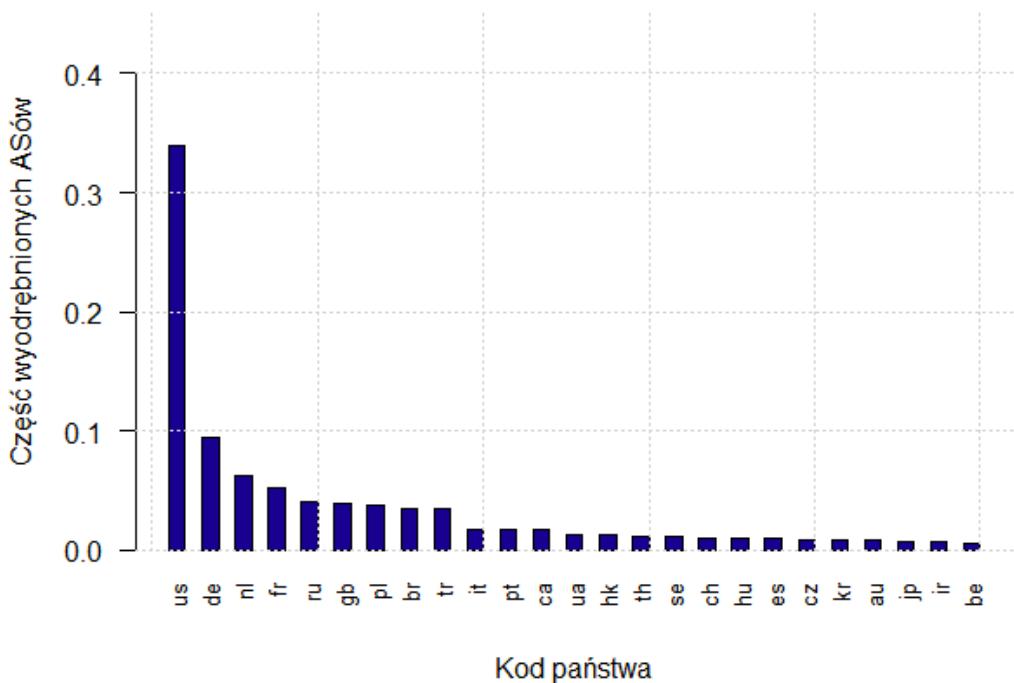


Rysunek 4.2: Zasada działania narzędzia mapującego adresy IP na numery AS.

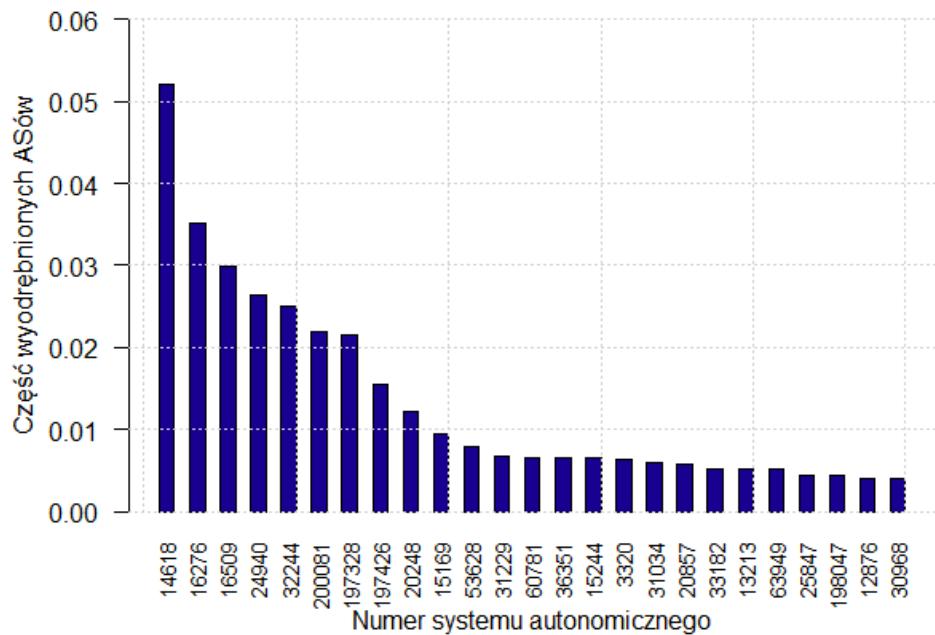
Jest to bardziej dokładna forma mapowania adresów IP na numery systemów autonomicznych. Translacja uzależniona jest od bazy danych, której wersję określa użytkownik. API, które zostało opisane na początku tego rozdziału jest całym komponentem, na który użytkownik nie ma wpływu. Dodatkowo, w przypadku poprzedniego systemu, do interfejsu serwera nie jest przekazywana data, więc należy zakładać, że adresy IP są mapowane na numery systemów autonomicznych na podstawie najnowszej bazy danych. Aby odwzorowanie było jak najbardziej wiarygodne, mapowanie adresów powinno być realizowane już w dniu przeprowadzania skanowania, co dodatkowo obciążałoby serwer i przełożyło się na niższą wydajność.

#### 4.2.3 Zbiór adresów IPv4

Analizując zbiór pod kątem adresów IP jedynie w czwartej wersji, zaprezentowany na rysunku 4.3, możemy zauważać kilka interesujących faktów. Pierwszym, który jest natychmiast zauważalny, to bardzo wysoki współczynnik systemów autonomicznych ze Stanów Zjednoczonych w populacji adresów IPv4 które zostały zebrane w wyniku skanowania. Poziom ten jest około 3.5 raza wyższy, niż liczliwość systemu autonomicznego z drugiego kraju w rankingu, czyli z Niemiec. Informacja ta jest interesująca w tym sensie, że TLD .de jest jedną z tych, które wymagają zabezpieczania stref DNS przed *zone enumeration* [11, 76]. Okazuje się natomiast, że w praktyce drugim najczęściej spotykanym krajem w zestawieniu adresów IP w skanowaniu AXFR są właśnie Niemcy, a transfer strefy jest w realizacji dużo łatwiejszym atakiem niż wspomniane wcześniej *zone walking* 2.5.



Rysunek 4.3: Pochodzenie geograficzne adresów IPv4 z podziałem na państwa. Adresy określone na podstawie numerów AS.



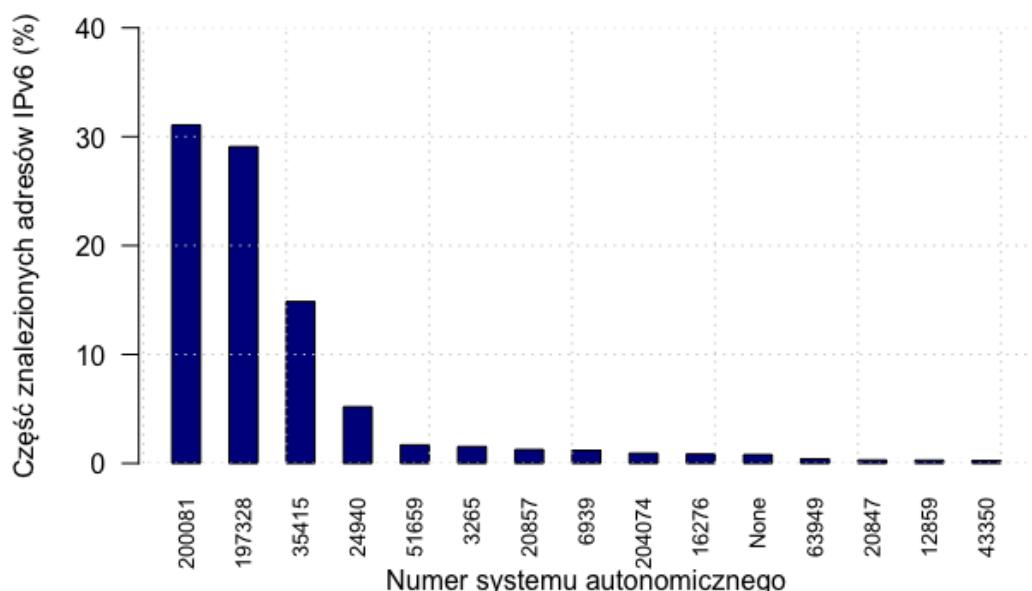
Rysunek 4.4: Zestawienie systemów autonomicznych, z których domeny najczęściej odpowiadają na zapytania AXFR.

Dodatkowo, zastanawiające jest, że oprócz bardzo widocznej przewagi systemów autonomicznych ze Stanów Zjednoczonych, widoczna jest również duża różnica pomiędzy 9 pierwszymi krajami oraz resztą populacji (etykieta *tr* (Turcja) na wykresie 4.3).

Analizując wykres 4.4 można dojść do wniosku, że istnieje kilka systemów autonomicznych, które zdecydowanie przeważają w liczbie adresów IPv4, które udało się uzyskać dzięki transferowi strefy AXFR. Możliwe, że administratorzy tych domen wiedzą o tym i specjalnie udostępniają taką możliwość.

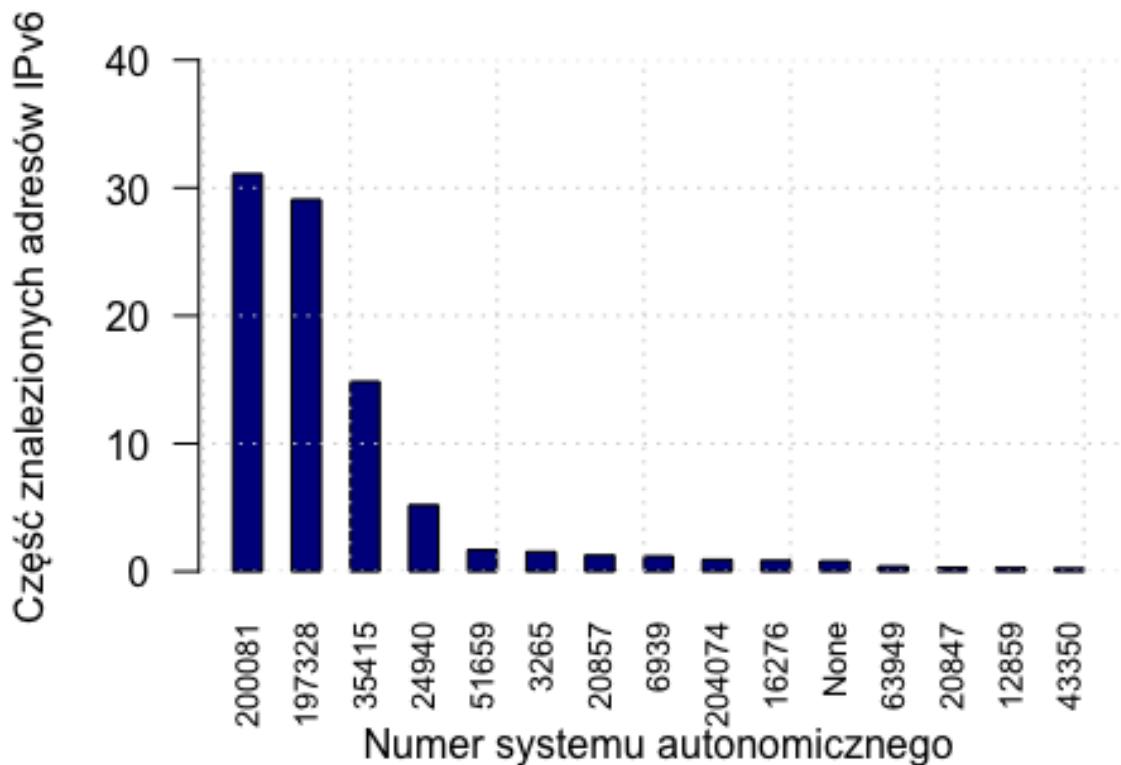
#### 4.2.4 Zbiór adresów IPv6

Oprócz opisanego w poprzednim podpunkcie 4.2.3 przeanalizowano zebrane dane pod względem zawartości adresów IP w wersji 6. Rozpatrując przypadek adresów IPv6 widzimy na rysunku 4.5 wyraźną przewagę systemów autonomicznych z trzech państw – Niemiec, Turcji oraz Holandii. Z pewnością wpływ na zaprezentowane wyniki miał również stopień zaawansowania wdrożenia IPv6 w poszczególnych państwach. Zastanawiająca jest stosunkowo niska pozycja systemów autonomicznych ze Stanów Zjednoczonych, jednak można przypuszczać, że jest to właśnie konsekwencja stanu wdrożenia protokołu IPv6. Kolejne z państw przedstawione na wykresie 4.5 stanowią już bardzo małą część wszystkich systemów autonomicznych – nawet poniżej 1%.



Rysunek 4.5: Pochodzenie geograficzne adresów IPv6 z podziałem na państwa. Adresy określane na podstawie numerów AS.

Analogicznie jak w przypadku opisany w rozdziale 4.2.3 zaprezentowano bardziej szczegółowy wykres, na podstawie którego możliwa jest analiza adresów ze względu na numery poszczególnych systemów autonomicznych. Widać dość wyraźną różnicę pomiędzy dwoma najbardziej licznymi zbiorami AS, a całą resztą znalezionych adresów. Co więcej, są to systemy autonomiczne stosunkowo młode, na co wskazują ich wysokie indeksy (200081 i 197328).



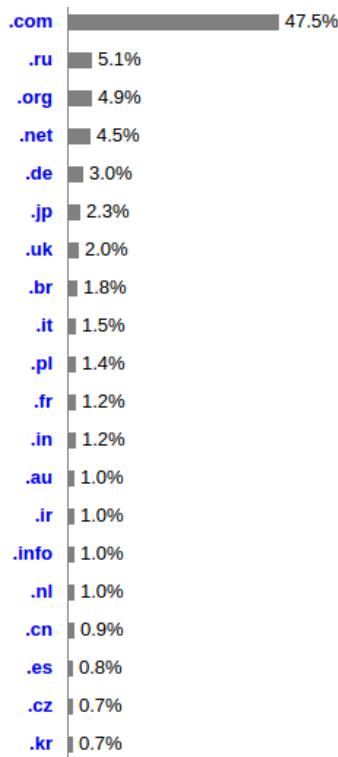
Rysunek 4.6: Pochodzenie geograficzne adresów IPv6 z podziałem na państwa. Adresy określane na podstawie numerów AS.

### 4.3 Analiza TLD

Jednym z podejść analizy zebranego zbioru danych było sprawdzenie, jak przedstawia się rozkład popularności domeny najwyższego poziomu dla serwerów podatnych na nieuprawniony transfer strefy.

W głównej mierze należy zastanowić się, czy rozkład ten jest w ogóle istotny, czy może nie różni się on niczym od ogólnego, całosciowego rozkładu popularności TLD w internecie, bez

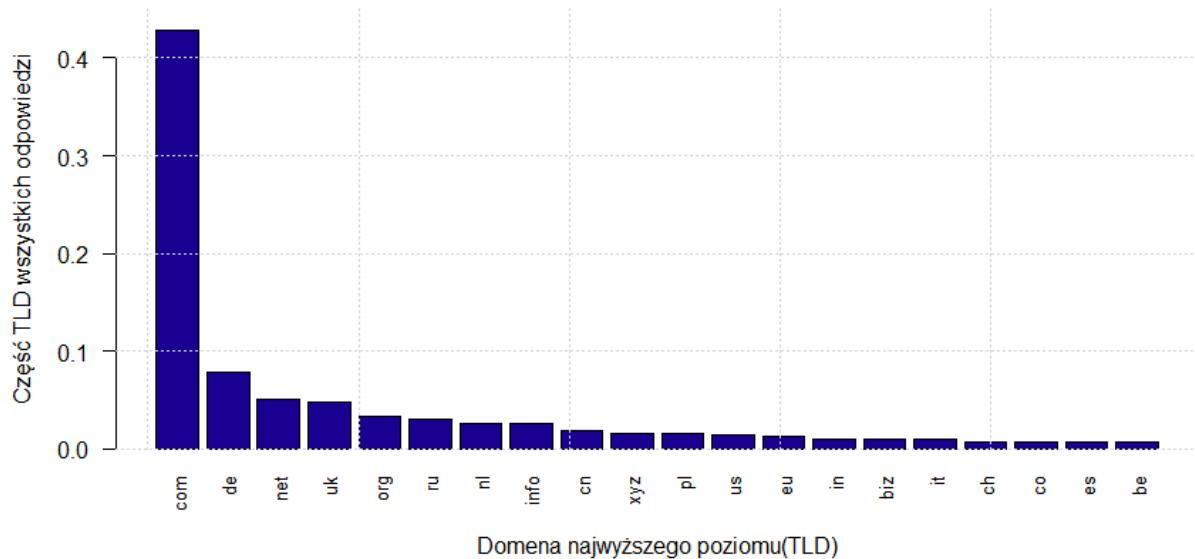
uwzględnienia podatności, czy innych czynników. W tym celu przeanalizowano i przedstawiono wykres popularności domen najwyższego poziomu w sieci Internet [91]. Jest on przedstawiony na rysunku 4.7.



Rysunek 4.7: Popularność TLD w internecie (dostęp na dzień 15. maja 2017), źródło: [91].

Wykres przedstawia procentowy rozkład poszczególnych domen internetowych w zależności od używanej domeny najwyższego poziomu. Wynika z niego, że domena najwyższego poziomu *.com* jest używana przez 47.5% wszystkich domen. Wykres ograniczony został do 20 najbardziej popularnych TLD. Różnice pomiędzy kolejnymi rekordami są na tyle niskie, że ich umieszczenie wpływałoby negatywnie na ogólną reprezentację wyników. Wykres 4.7 zostanie następnie zestawiony z popularnością TLD tych domen, których serwery autorytatywne umożliwiają nieuprawniony transfer AXFR. Rozkład popularności TLD domen odpowiadających na zapytanie AXFR przedstawiono na rysunku 4.8.

Dokonując porównania wykresów 4.7 oraz 4.8 możemy dostrzec kilka prawidłowości. Zarówno w jednym jak i drugim przypadku, wyraźnie dominującą domeną najwyższego poziomu jest domena *.com*. Dodatkowo, nie można mówić tu o anomalii, ponieważ zarówno w jednym jak i drugim przypadku domena *.com* stanowi bardzo podobny procentowy udział wszystkich domen, czy



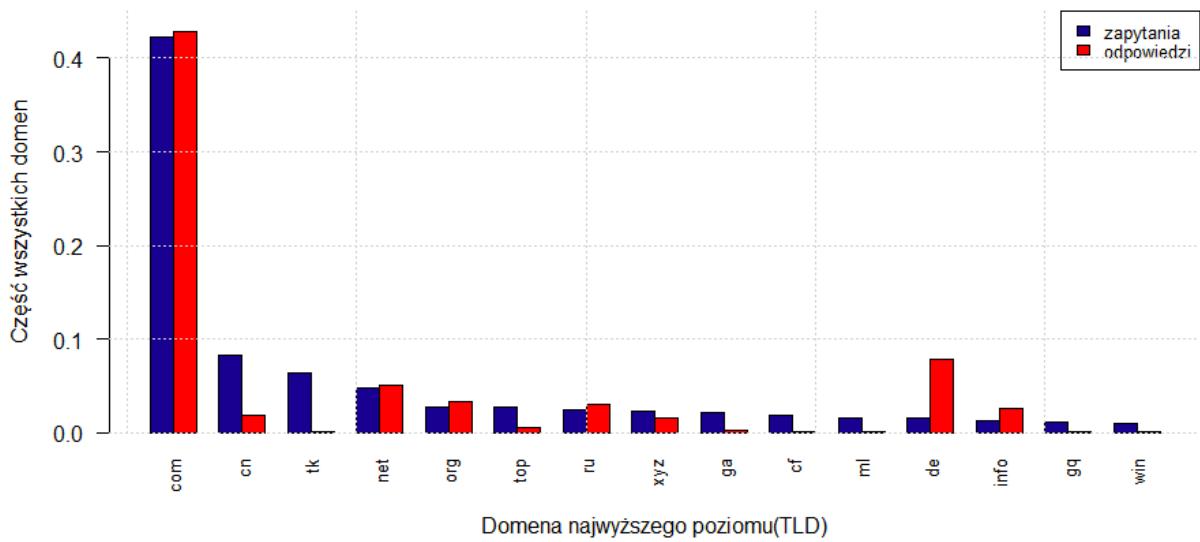
Rysunek 4.8: Popularność TLD w odpowiedziach AXFR.

to pytanych czy tych, które odpowiedziały. Następnie można dostrzec delikatne zmiany w procentowych udziałach kolejnych TLD, jednak nie na pierwszy rzut oka zmiany te nie są wyróżniające się czy nad wyraz zauważalne.

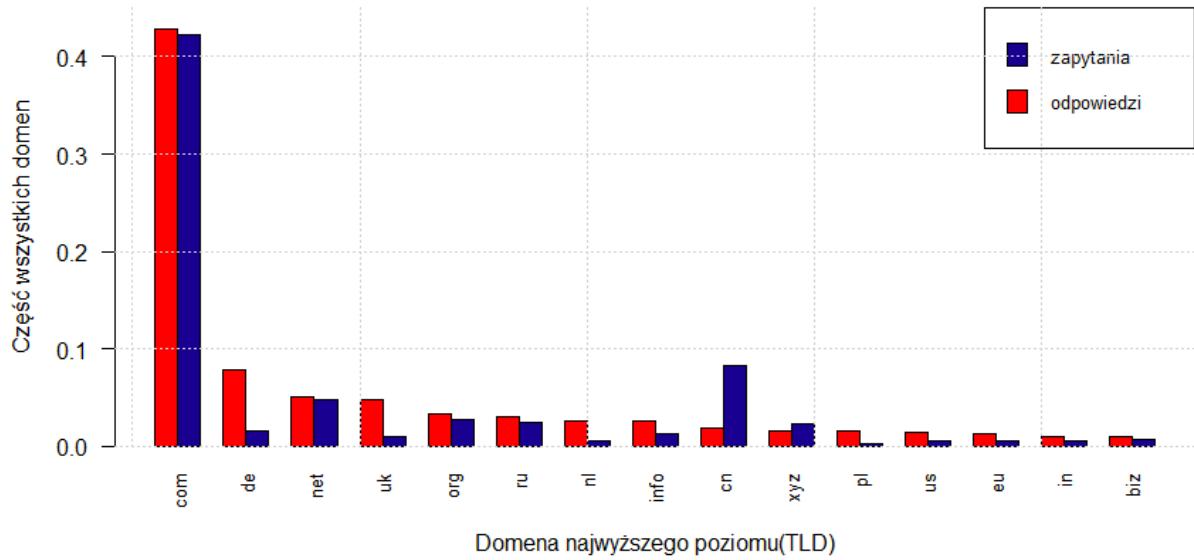
W związku z tym, że wyżej opisane działania nie pozwoliły na wyraźne zarysowanie różnic pomiędzy rozkładami TLD, zdecydowano się zestawić ze sobą inne zbiory danych. Wykreślono podobne wykresy dla popularności domeny najwyższego poziomu dla domen odpytywanych podczas skanowania oraz TLD domen, które na zapytania odpowiedziały. Następnie zestawiono ze sobą wyniki pozyskane z obu tych zbiorów danych. Zostały odpowiednio wykreślone:

1. wykres 15 najpopularniejszych TLD w zbiorze domen odpytywanych w porównaniu do popularności tych TLD w zbiorze domen, które odpowiedziały na zapytanie AXFR – wykres 4.9,
2. wykres 15 najpopularniejszych TLD w zbiorze domen, które odpowiedziały na zapytanie AXFR w zestawieniu z ich popularnością w puli TLD domen odpytywanych – wykres 4.10.

W przypadku tych bezpośrednich zestawień, można zauważyć więcej interesujących prawidłowości. Na początku warto wspomnieć, że istnieje duża dysproporcja w przypadku niektórych domen najwyższego poziomu. Pierwszą z nich może być przypadek domeny .de, gdzie procentowy udział TLD .de we wszystkich TLD odpowiedzi AXFR jest drugim wynikiem podczas, gdy w populacji danych wejściowych jest to 12 najliczniejsza grupa. Warto dodać, że domena .de jest jedną z domen, których zarządcy traktują proceder *zone enumeration* jako łamanie prawa, co opisano w podpunkcie 2.5. Transfer AXFR pozwala na uzyskanie danych o podobnej charakterystyce, a mimo to, ok 8% wszystkich odpowiedzi powiązane jest z domeną najwyższego poziomu .de



Rysunek 4.9: Zestawienie odpowiedzi ze względu na rozkład danych wejściowych. Rozkład odpowiedzi znormalizowano do całkowitej liczby odpowiedzi, zaś rozkład zapytań do całkowitej ich liczby.



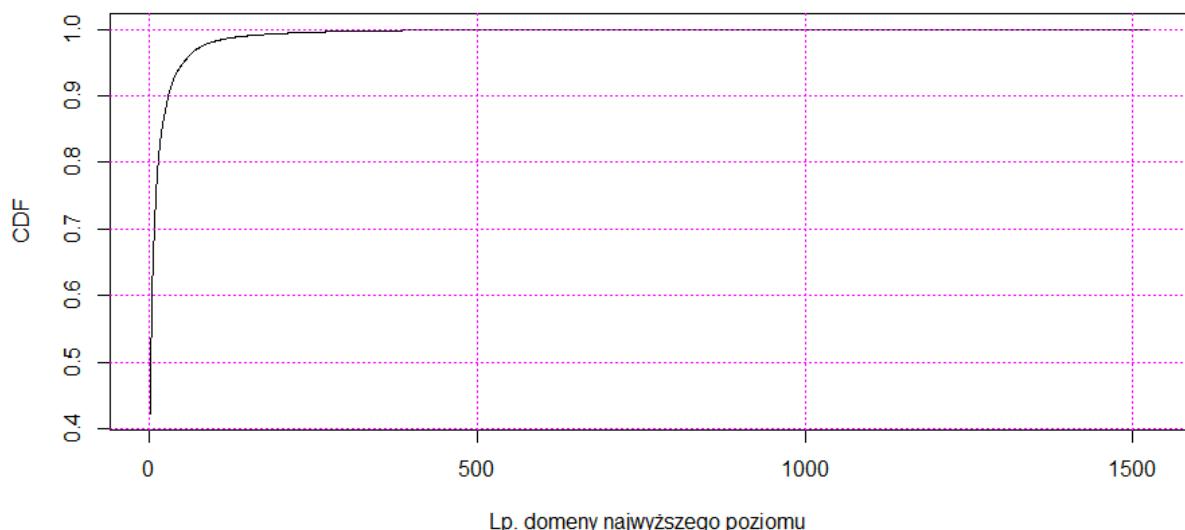
Rysunek 4.10: Zestawienie częstości występowania domen najwyższego poziomu dla domen, dla których możliwy jest AXFR z ich populacją w zbiorze danych wejściowych. Rozkład odpowiedzi znormalizowano do całkowitej liczby odpowiedzi, zaś rozkład zapytań do całkowitej ich liczby.

(domena krajowa – Niemcy). Podobna sytuacja ma miejsce również w przypadku TLD .uk (Wielka Brytania) czy .pl (Polska), gdzie przy małym współczynniku domen odpytywanych otrzymano nieproporcjonalnie wysoki współczynnik odpowiedzi.

Zjawiskiem w pewnym stopniu odwrotnym do opisanego w poprzednim akapicie jest rozkład

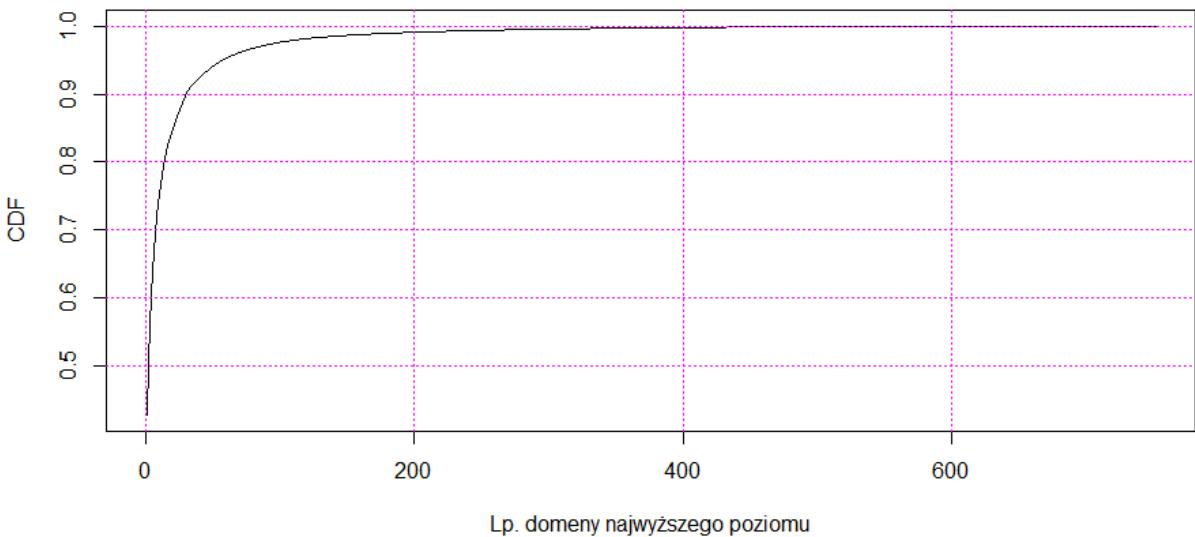
zapytań/odpowiedzi dla TLD .cn (Chiny) lub .tk (Tokelau – region zależny od Nowej Zelandii). W tym przypadku zauważalna jest również duża dysproporcja, jednak „na korzyść”, gdyż w stosunku do wielu zapytań wystosowanych do domen należących do tych TLD uzyskano relatywnie mało odpowiedzi. Domena .tk jest interesującym przykładem również z innego względu. Jej operator – DotTK umożliwia rejestrację darmowych domen. W związku z taką polityką, TLD .tk stała się bardzo popularna wśród cyberprzestępcołów. W 2007 roku firma McAfee uznała ją za najbardziej niebezpieczną spośród wszystkich domen [79]. W 2010 roku domena ta znalazła się znacznie niżej w rankingu [80], jednak wciąż była to wysoka, jedenasta pozycja. W obu raportach domena .cn również znajduje się na wysokich miejscach. Również TLD .top można zaliczyć do domen o niskiej proporcji zapytań względem odpowiedzi. Została ona oficjalnie wydzielona w 2014 roku. Jej operatorem jest firma „top registry” znajdująca się w Chinach, a domeny zarejestrowane w TLD .top muszą używać chińskich serwerów.

Oprócz opisanych wcześniej względnych porównań pomiędzy pytaniami i odpowiedziami od serwerów autorytatywnych różnych domen sprawdzono jak wygląda dystrybuanta poszczególnych rozkładów. Pozwoliło to oszacować ile (ilościowo) domen najwyższego poziomu zawierało się w grupie, dla której generowano większość zapytań (rysunek 4.11). Analogicznie, w przypadku danych dotyczących odpowiedzi AXFR można było ocenić ile różnych TLD pojawią się niemal we wszystkich odpowiedziach (rysunek 4.12).



Rysunek 4.11: Dystrybuanta domeny najwyższego poziomu w odpytywanych domenach.

Oprócz wykreślonych charakterystyk policzono również ile domen najwyższego poziomu gromadzi w sobie 99% zapytań bądź odpowiedzi (w zależności od badanego zbioru). Wyniki zaprezentowano w tabeli 4.3. Warto zauważyć, że pomimo ponad dwukrotnie mniejszego zbioru różnych domen poziomu najwyższego, należy zgrupować dużo więcej TLD, aby uzyskać zbiór 99% wszystkich odpowiedzi. Można z tego wnioskować, że rozkład odpowiedzi jest bardziej jednostajny niż



Rysunek 4.12: Dystrybuanta domeny najwyższego poziomu w zbiorze domen odpowiadających strefą DNS na zapytanie AXFR.

w przypadku danych wejściowych. Można także przypuszczać, że zbiór wejściowy zawierał pojedyncze domeny z różnych, rzadko spotykanych TLD. W momencie, kiedy jedna, czy jedynie kilka domen z takiego TLD są odpowiednio zabezpieczone przed nieuprawnionym transferem AXFR nie znajdziemy danego sufiksu w zbiorze odpowiedzi. Sytuacja taka może mieć miejsce, gdy wszystkie domeny z mało popularnego TLD znajdują się pod zarządem jednej osoby czy organizacji.

Zbiór	Liczba różnych TLD w zbiorze	Liczba różnych TLD zawierająca 99% elementów zbioru
Zapytania	1521	147
Odpowiedzi	751	187

Tabela 4.1: Opis zbiorów zapytań/odpowiedzi

#### 4.4 Liczba wpisów stref w odpowiedziach

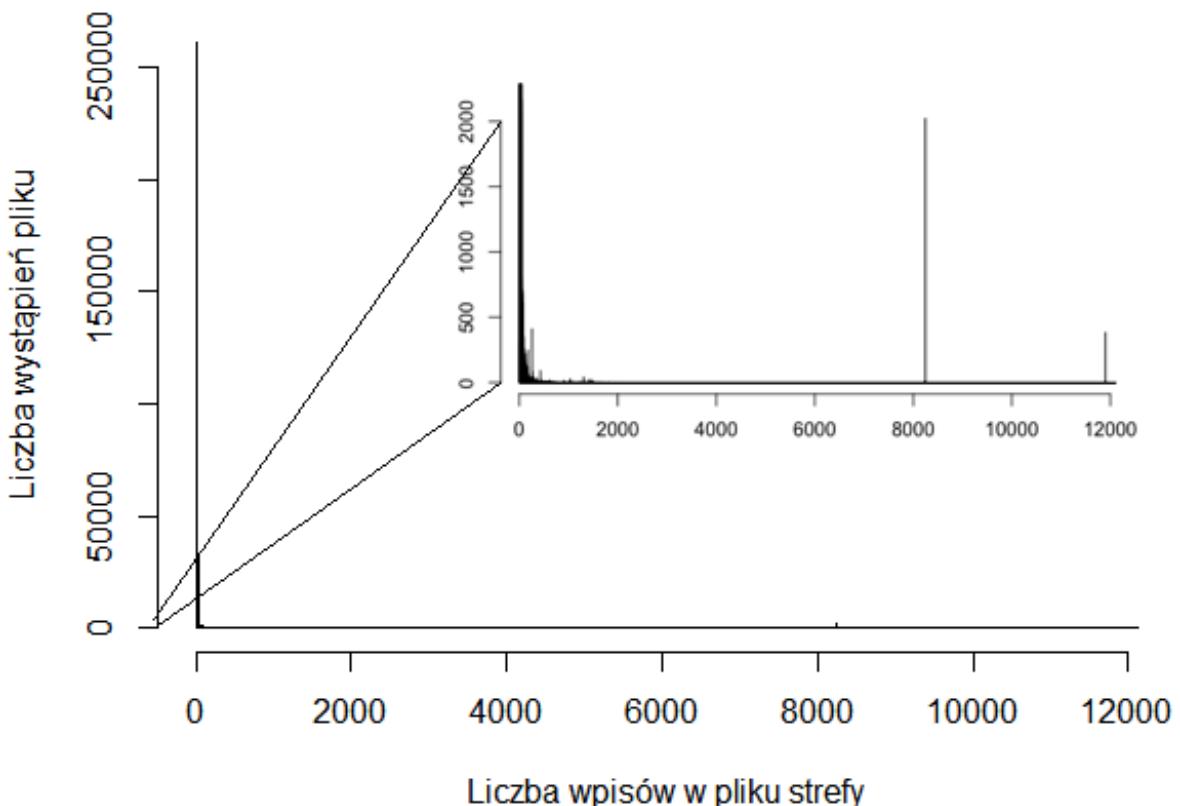
Odpowiedzi pobrane w wyniku skanowania zostały poddane analizie uwzględniającej jak duże są strefy pobrane z serwerów przestrzeni nazw.

W tym przypadku zastosowano inne podejście niż podczas reszty przeprowadzonych badań. Z racji tego, że częstym przypadkiem odpowiedzi otrzymywanej od serwera był jedynie rekord SOA/CNAME, postanowiono oddzielnie analizować pliki stref zawierające tylko jeden wpis, a oddzielnie pliki zawierające więcej niż jeden rekord.

W wyniku przeprowadzonych badań ustalono, że:

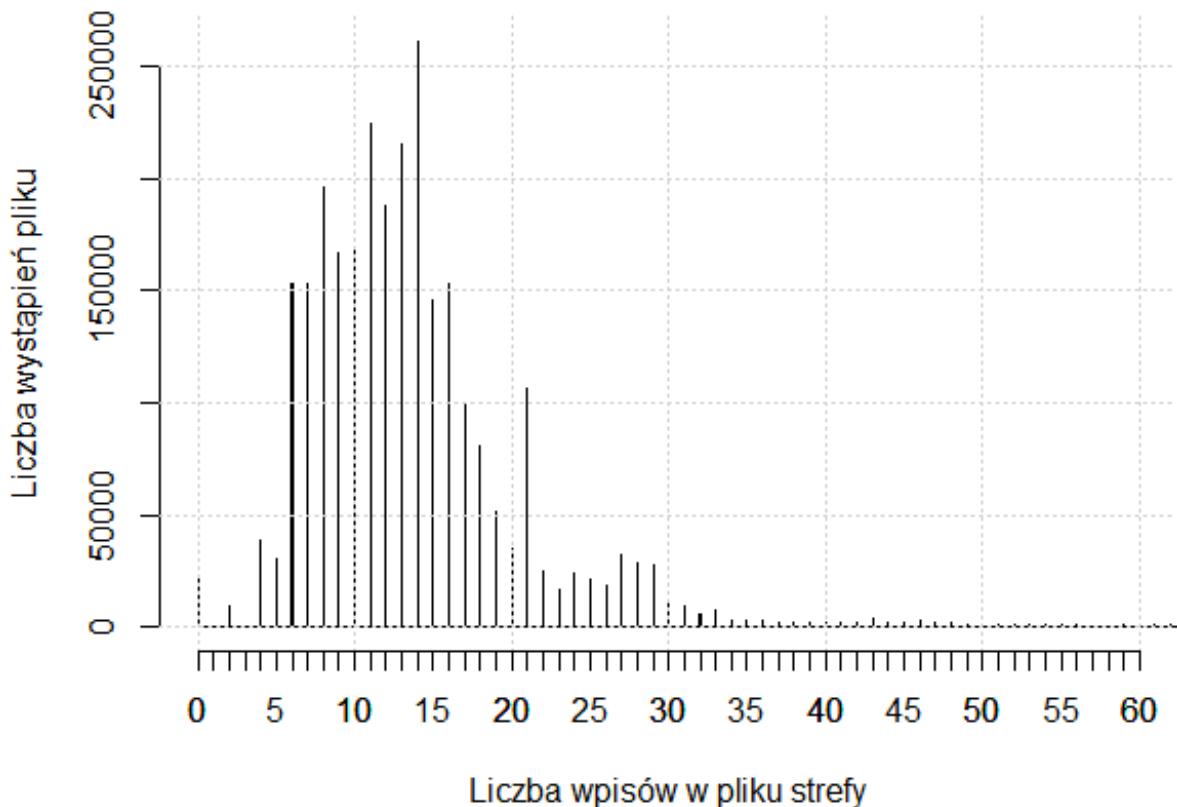
1. 6179845 z 8985154 (68,8%) plików stref zawiera jedynie jeden wpis (rekord SOA bądź rekord CNAME),
2. 2803902 z 8985154 (31,2%) plików stref zawiera więcej niż jeden wpis.

Przypadek, gdy plik strefy zawiera więcej niż jeden wpis można utożsamić ze źle skonfigurowanym serwerem autorytatywnym domeny. Zwracanie samego wpisu SOA (czyli odpowiednik zbioru danych z jednym wpisem w pliku strefy) nie może być traktowane jako błąd, gdyż w dokładnie taki sam sposób zachowuje się serwer DNS w przypadku podania błędnego numeru serii w transferze IXFR. Tak zaprojektowany został inkrementalny transfer strefy DNS zaprezentowany w RFC1995 [70] i do tej pory nie zaprezentowano ewentualnych podatności, które mogłyby wykorzystywać ten protokół.



Rysunek 4.13: Rozkład liczby rekordów w odpowiedziach AXFR.

Dla zbioru danych, który określa rozmiar pobranych stref wykreślony został histogram, z którego można odczytać, jak często spotykane są strefy DNS o określonych rozmiarach. W przypadku skanów polegających na transferze stref widoczna jest bardzo duża częstość występowania domen o wpisach do kilkudziesięciu rekordów. Pierwszy z histogramów na których zaprezentowano opisane częstotliwości (rysunek 4.13) sugeruje, że dużo bardziej interesująca jest pierwsza część całego



Rysunek 4.14: Rozkład liczby rekordów w odpowiedziach AXFR, widok powiększony.

wykresu. Zaprezentowano ją, w powiększeniu, na kolejnym rysunku 4.14. Dodatkowo, w górnej części rysunku 4.13 umieszczono histogram w powiększeniu, z bardzo ograniczoną skalą na osi rzędnych. Pozwala to, na zaobserwowanie prążków, które nie były widoczne na histogramie prezentującym cały zbiór.

Dodatkowo, można zauważyć, że histogram 4.13 posiada bardzo dużo pozycji na osi X. Jest to ponownie następstwo późniejszej implementacji oprogramowania umożliwiającego transfer stref AXFR. Zauważalny jest prążek na wartości około 8 tysięcy wpisów. Dokładnie jest to 8244 wpisów w pliku strefy. Sprawdzono, czy rzeczywiście strefy te są tak duże. Odpytano kilka z serwerów, dla których otrzymano tak specyficzną odpowiedź. Pakiety, które wysłano oraz odebrane zostały przeanalizowane programem Wireshark. Okazuje się, że utworzenie opisanego wcześniej pliku jest następstwem otrzymania wiadomości HTTP o statusie 400 – Bad request. Co ciekawe, odpowiedź tę otrzymujemy na mało standardowy port w kontekście protokołu HTTP – port 53. Szczególnie otrzymanej wiadomości zostały przedstawione w listingu 4.1. Innym wynikiem typu zachowania były odpowiedzi o liczbie linii ok. 11 tysięcy. Jest to wyraźnie mniej zauważalne, wnioskując choćby po histogramie 4.13, gdzie widoczny jest jedynie prążek na wartości około 8000 linijek w pliku strefy. Plik o około 11 tysiącach (11900 linii) wpisów był tworzony w wyniku otrzymania

rozszerzonej, głównie o tagi HTML wiadomości HTTP 400 – została ona przedstawiona na listingu 4.2.

---

```
1 HTTP/1.0 400 Bad request
2 Cache-Control: no-cache
3 Connection: close
4 Content-Type: text/html
5
6 <html><body><h1>400 Bad request</h1>
7 Your browser sent an invalid request.
8 </body></html>
```

---

Listing 4.1: Odpowiedź HTTP 400 na zapytanie DNS.

---

```
1 <html>
2 <head><title>400 Bad Request</title></head>
3 <body bgcolor="white" >
4 <center><h1>400 Bad Request</h1></center>
5 <hr><center>nginx/1.4.0</center>
6 </body>
7 </html>
```

---

Listing 4.2: Rozszerzona odpowiedź HTTP 400 na zapytanie DNS.

Na rysunku 4.14 możemy obserwować w przybliżeniu rozkład wielkości plików stref. Zawarty został on do 60 wpisów z uwagi, że najbardziej interesujące wyniki zawierają się głównie w tym przedziale. Jak widać, najczęściej występującym rozmiarem strefy DNS jest 14 wpisów. Są to strefy pobrane najczęściej ze źle skonfigurowanych serwerów DNS, które obsługują usługi standardowe, często spotykane, takie jak serwery WWW, serwer pocztowe. Przykład odpowiedzi o takim rozmiarze został przedstawiony na listingu 4.3.

---

```
1 movida-bar.pl. 14400 6 ns1.consultingteam.pl. hostmaster.←
    movida-bar.pl. 1952801133 3222735464 1869837421 1634956389 ←
    1925188727
2 movida-bar.pl. 14400 15 mail.movida-bar.pl
3 movida-bar.pl. 14400 16 v=spf1 a mx ip4:91.241.61.119 ~all
4 movida-bar.pl. 14400 1 91.241.61.119
5 movida-bar.pl. 14400 2 ns1.consultingteam.pl
6 movida-bar.pl. 14400 2 ns2.consultingteam.pl
7 ftp.movida-bar.pl. 14400 1 91.241.61.119
8 localhost.movida-bar.pl. 14400 28 ←
```

```

0000:0000:0000:0000:0000:0000:0001
9 localhost.movida-bar.pl.      14400   1      127.0.0.1
10 mail.movida-bar.pl.        14400   1      91.241.61.119
11 pop.movida-bar.pl.        14400   1      91.241.61.119
12 smtp.movida-bar.pl.       14400   1      91.241.61.119
13 www.movida-bar.pl.        14400   1      91.241.61.119
14 movida-bar.pl.    14400   6      ns1.consultingteam.pl. hostmaster.←
                           movida-bar.pl. 2013010900 14400 3600 1209600 86400

```

---

Listing 4.3: Przykładowa odpowiedź na żądanie strefy DNS.

Z listingu 4.3 możemy dowiedzieć się, że domena obsługiwana przez odpytany serwer DNS udostępnia między innymi takie usługi jak serwery pocztowe smtp i pop, serwer FTP oraz serwer WWW. Wszystkie one są kierowane na jeden adres IP. Dodatkowo, na serwerze DNS został umieszczony wpis TXT (id 16). Zdefiniowano w nim rekord SPF gdzie określa się reguły dotyczące wymiany wiadomości e-mail w danej domenie. Odwzorowanie indeksów poszczególnych rekordów DNS można odczytać z tabeli 3.2.

Strefy tego typu są najczęściej spotykane w danych, które udało się zebrać podczas skanowania. Można wnioskować, że transfer AXFR jest najczęściej możliwy dla domen zarejestrowanych głównie w celach informacyjnych czy marketingowych. Jak opisano wcześniej, są to najczęściej pojedyncze adresy IP z kilkoma usługami. Działalność organizacji, na potrzeby których rejestrowano takie niewielkie domeny, jest najczęściej mało związana z informatyką czy telekomunikacją.

To, na co warto zwrócić uwagę na histogramie 4.14 to również domeny o strefach o rozmiarze około 40-60 wpisów. Jest to reprezentacja tych stref, które wykorzystują mechanizm podpisywania rekordów DNS zaprezentowany w rozszerzeniu protokołu – DNSSEC [15, 14]. Zgodnie z tym, co zostało opisane w rozdziale 1.4.1. Dla każdej pary rekordów przechowywanych w strefie DNS liczony jest podpis cyfrowy. Serwery, które odpowiedziały strefą zawierającą podpisy kryptograficzne najprawdopodobniej wspierają DNSSEC, co może zostać użyte przez cyberprzestępco&w;w do

## 4.5 Odpowiedzi niestandardowe

Jak wspomniano w podpunkcie 4.1, wyjątkową sytuacją, którą udało się zaobserwować, były odpowiedzi o nietypowym rozmiarze 25123 bajtów. Przykładowa komunikacja z serwerem odpowiadającym w ten sposób została zaprezentowana na listingu 4.4.

```

1 mskwarek\$ dig tnttelivision.com @173.255.218.70 axfr
2
3 ; <>> DiG 9.8.3-P1 <>> tnttelivision.com @173.255.218.70 axfr
4 ;; global options: +cmd

```

```

5 tnttelivision.com. 86400 IN SOA ns1.parklogic.com. hostmaster.←
    tnttelivision.com. 2017061500 16384 2048 1048576 2560
6 ;; communications error to 173.255.218.70#53: end of file

```

Listing 4.4: Przykładowy odpowiedź serwera.

W tym samym czasie ustalono nasłuchiwanie programu Wireshark [87] na odpowiednim interfejsie oraz odfiltrowano ruch sieciowy nie związany z komunikacją z serwerem przestrzeni nazw o adresie 173.255.218.70. Ogólny przebieg komunikacji można zobaczyć na zrzucie ekranu 4.15.

No.	Time	Source	Destination	Protocol	Length	Info
2749	163.046964	192.168.43.101	173.255.218.70	TCP	78	53543 → 53 [SYN] Seq=0 Win=65535 Len=0 MSS=14...
2750	163.447696	173.255.218.70	192.168.43.101	TCP	74	53 → 53543 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0...
2751	163.447786	192.168.43.101	173.255.218.70	TCP	66	53543 → 53 [ACK] Seq=1 Ack=1 Win=131200 Len=0...
2752	163.449028	192.168.43.101	173.255.218.70	DNS	103	Standard query 0xccf0 AXFR tnttelivision.com
2756	163.746008	173.255.218.70	192.168.43.101	TCP	66	[TCP Previous segment not captured] 53 → 5354...
2757	163.746014	173.255.218.70	192.168.43.101	TCP	66	53 → 53543 [ACK] Seq=1 Ack=38 Win=29056 Len=0...
2758	163.746015	173.255.218.70	192.168.43.101	TCP	178	[TCP Out-Of-Order] 53 → 53543 [PSH, ACK] Seq=...
2759	163.746117	192.168.43.101	173.255.218.70	TCP	78	[TCP Dup ACK 2751#1] 53543 → 53 [ACK] Seq=38 ...
2760	163.746118	192.168.43.101	173.255.218.70	TCP	66	53543 → 53 [ACK] Seq=38 Ack=114 Win=131104 Len=...
2765	163.756421	192.168.43.101	173.255.218.70	TCP	66	53543 → 53 [FIN, ACK] Seq=38 Ack=114 Win=1311...
2778	164.034097	173.255.218.70	192.168.43.101	TCP	66	53 → 53543 [ACK] Seq=114 Ack=39 Win=29056 Len=...

Rysunek 4.15: Przebieg komunikacji z serwerem wysyłającym pakiet TCP z flagą RST.

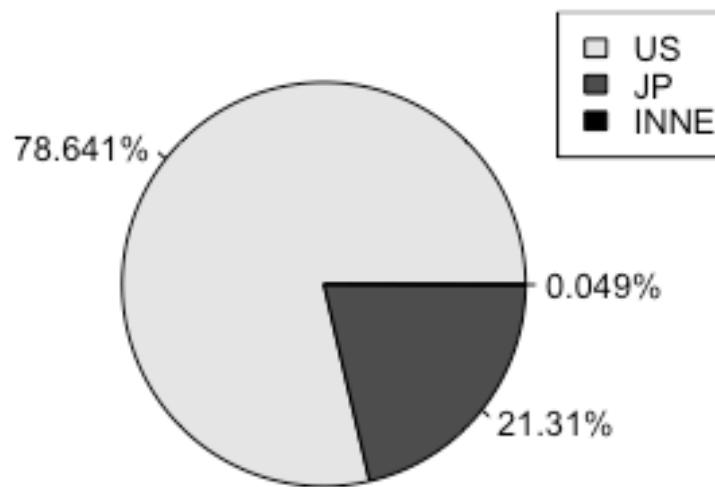
Postanowiono sprawdzić, czy domeny oraz serwery autorytatywne powiązane z takimi przypadkami mają wspólne cechy. Pierwszym kryterium porównania wspomnianych przypadków była analiza systemów autonomicznych, do których przynależą dane serwery przestrzeni nazwy. Wyniki zaprezentowano w tabeli 4.5.

Niewątpliwie interesującym zjawiskiem jest fakt, że znaczna większość tych specyficznych plików pochodzi z kilku ASów. Może to sugerować, że właśnie w tych systemach autonomicznych operatorzy wykorzystują oprogramowanie specjalnie modyfikowane pod swoje potrzeby, jednak nie zawsze zgodne ze specyfikacją ustalaną w kolejnych dokumentach RFC. Wynik zapytania, który zaprezentowano na listingu 4.4 zawiera rekord SOA, który wskazuje, że podstawowym serwerem dla tej domeny jest serwer *ns1.parklogic.com*. Nazwa tego serwera może sugerować, że dana domena została zaparkowana. To, że domeny, dla których zwracany jest tylko jeden wpis są zaparkowane może być dobrym uzasadnieniem niewielkiej różnorodności państw, numerów systemów autonomicznych czy domen najwyższego poziomu dla tego zbioru. Domena, którą zaparkowano w danej firmie obsługiwana jest przez jej serwery. Firmy oferujące takie usługi są często bardzo duże [45], rynek parkowania domen jest więc oligopolem, co potwierdzają otrzymane wyniki.

Na wykresie 4.16 pokazano procentowy rozkład krajów z których pochodziły serwery przestrzeni nazw. Jak widać, serwery z jedynie dwóch krajów (Japonii oraz Stanów Zjednoczonych) stanowią liczną grupę w tym zestawieniu. Pozostałe kraje, nawet zebrane we wspólną grupę stanowią zaledwie 0,5%.

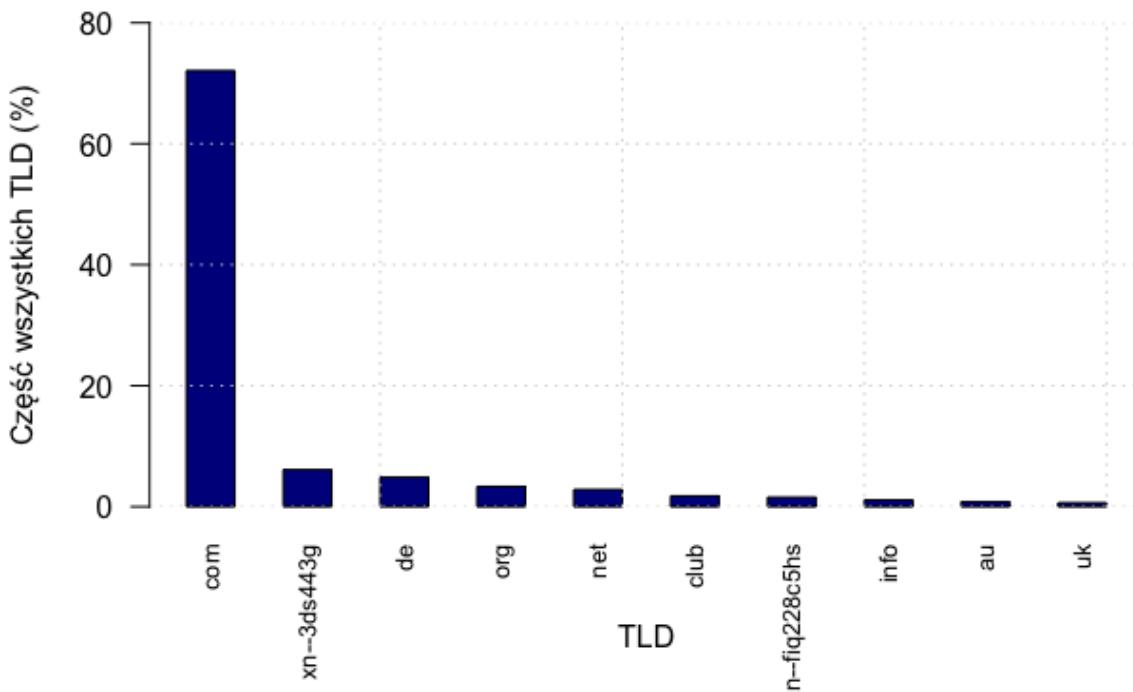
Numer AS	Nazwa AS	Liczba wystąpień
63949	LINODE-APLinode, LLC, US	324483
32181	ASN-GIGENET-GigeNET, US	168263
2516	KDDI KDDI CORPORATION, JP	133523
4837	CN	290
29676	GRADWELL, GB	8
263632	BR	6
46844	ST-BGP-Sharktech, US	2
44682	TELECOMROMANIA Strada Peroninr.45, RO	2
12876	AS12876, FR	2
701	UUNET-MCI Communications Services, Inc. d/b/a VerizonBusiness, US	1
45413	SERVENET-AS-TH-AP Serve NET Solution Limited Partnership, TH	1
12374	LFNET-AS01, DE	1

Tabela 4.2: Tabela liczebności AS w opisywanym przypadku



Rysunek 4.16: Lokalizacja serwerów nazw odpowiadających pakietem TCP z flagą RST.

Przedstawiona proporcja jest ciekawa w kontekście analizy domen najwyższego poziomu, które obsługują serwery nazw opisane w poprzednim akapicie. Domeny znajdujące się w TLD .com stanowią 75% wszystkich domen tego typu, co znacznie przewyższa średni udział TLD .com w ogólnej liczbie TLD (rysunek 4.7).



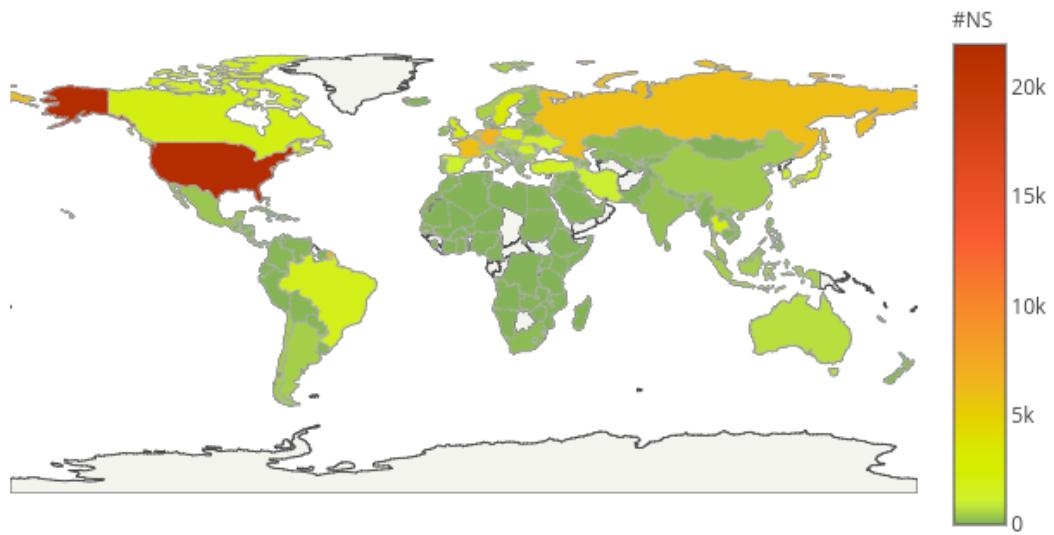
Rysunek 4.17: TLD domen, których serwery autorytatywne odpowiadają TCP RST.

## 4.6 Geograficzna lokalizacja serwerów

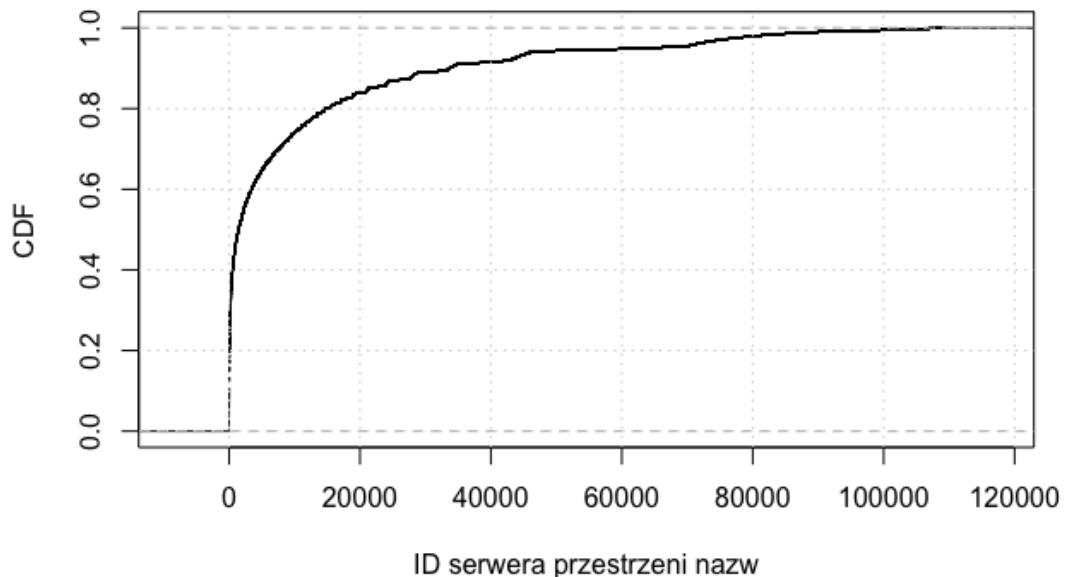
Biorąc pod uwagę poprzednie rozważania postanowiono, że warto przeanalizować geograficzne rozmieszczenie serwerów przestrzeni nazw, które odpowiedziały na zapytanie AXFR. Podział zaprezentowany w tym punkcie jest już bardziej szczegółowym rozbiciem konkretnych wyników. Najważniejszym kryterium podziału domen było przede wszystkim rozdzielenie plików stref oraz odpowiadających im serwerów przestrzeni nazw w zależności od tego jak duży plik strefy udało się dla nich pozyskać.

Pierwszym wykresem który zaprezentowano jest mapa adresów IP serwerów przestrzeni nazw, które odpowiadają plikiem strefy zawierającym więcej niż jeden rekord DNS. Należy dodać, że pliki „sztuczne”, takie, które powstały w wyniku niedoskonałości napisanego oprogramowania nie zostały uwzględnione podczas niniejszej analizy. Wyniki zaprezentowano na rysunku 4.18.

Zbiór adresów IP serwerów przestrzeni nazw który zaprezentowano na mapie 4.18 liczył 109160 unikalnych adresów, podczas gdy zbiór wszystkich plików stref danego typu dla analizowanego zbioru wynosił 2132448. Oznacza to, że na jeden serwer autorytatywny średnio przypada ponad 19 domen. Wykreślona została dystrybuanta rozkładu popularności serwerów przestrzeni nazw odpowiadających na zapytania AXFR. Przedstawiono ją na rysunku 4.19. Widoczna jest na nim ciekawa



Rysunek 4.18: Lokalizacja serwerów DNS, które odpowiadły więcej niż jednym rekordem.



Rysunek 4.19: Dystrybuanta rozkładu popularności serwerów przestrzeni nazw, które odpowiadają na zapytania AXFR.

tendencja, że odpowiedzi o połowie wszystkich domen, dla których pobrano informacje, pochodzą od relatywnie niewielkiej liczby serwerów przestrzeni nazw. Dokładne kwantyle przedstawionego

rozkładu zawarto w tabeli 4.4.

Z danych zawartych w tabeli 4.4 wynika, że 75% (trzeci kwantyl) wszystkich domen, które umożliwiają transfer strefy obsługiwanych jest przez jedynie 9,69% autorytatywnych serwerów przestrzeni nazw. Trend ten jest również zauważalny na wykresie dystrybuanty 4.19, gdzie od poziomu 0,75 na osi OY charakterystyka jest bardziej łagodna niż w pozostałej części wykresu. Wyniki te oznaczają, że dla pozostałych 25% domen autorytatywnych jest 90,21% wszystkich serwerów przestrzeni nazw. Najpopularniejsze adresy IP serwerów NS zapisano również w tabeli 4.3 wraz z liczbą domen, dla których dany serwer jest autorytatywnym.

Lp.	Adres IP	Liczba domen
1	167.160.13.69	16503
2	167.160.13.70	16318
3	162.243.45.236	14254
4	199.255.159.226	13735
5	198.204.224.146	13439
6	173.208.189.26	13222
7	209.160.33.82	13153
8	104.219.19.52	13113
9	192.151.149.92	12602
10	101.200.203.5	10659

Tabela 4.3: Autorytatywne NS dla największej liczby domen.

Kwantyl	Liczba serwerów	Procent wszystkich serwerów
I	136	0,12%
II	1447	1,33%
III	10579	9,69%
IV	109160	100%

Tabela 4.4: Kwantyle rozkładu popularności serwerów przestrzeni nazw, które odpowiadają na AXFR.

## 4.7 Usługi

Przeprowadzone skanowanie umożliwiło również sprawdzenie jakie usługi uruchamiane są wewnątrz organizacji, których serwery DNS były odpytywane. Zdecydowano się na sprawdzanie usług typowych dla branży ICT. Poszukiwano w zebranych danych informacji o serwerach systemów kontroli wersji (git, svn) oraz systemów automatycznego budowania oprogramowania i zapewniania ciągłej integracji (jenkins). W niniejszej pracy magisterskiej zdecydowano się głównie na te usługi ze względu na dużą wrażliwość takich danych jak kody źródłowe przechowywane na

serwerach git czy svn a także ze względu na istotę elementów środowiska pracy jak systemy CI typu Jenkins.

Dodatkowo sprawdzono jak często w odebranych strefach możliwe jest znalezienie informacji na temat usług pocztowych. W tym przypadku głównym celem poszukiwań były wpisy typu MX (id: 15) definiujące serwery wymiany poczty elektronicznej w danej domenie. Oprócz wcześniej wspomnianych usług, postanowiono także sprawdzić popularność serwerów wymiany plików ftp.

#### **4.7.1 Usługi wspomagające rozwój oprogramowania**

Jeśli chodzi o usługi, które zostały wymienione w pierwszym akapicie tego rozdziału, to na podstawie samego adresu URL udało się uzyskać informacje o ponad 3 tysiącach serwerach systemu git oraz ponad 3.5 tysiącach systemów svn. W odniesieniu do skali przeprowadzonych badań jest to nieduża liczba, co dobrze świadczy o społeczności. Niemniej jednak, liczba około 7 tysięcy systemów kontroli wersji, o których informacje uzyskano jedynie poprzez skanowanie AXFR jest wysoką liczbą. Szczególnie, jeśli uwzględniony zostanie fakt, że systemy te najczęściej wykorzystywane są przez osoby o wykształceniu technicznym, które są świadome istoty bezpieczeństwa sieciowego.

Kolejną z wymienionych usług był serwer CI (ang. *Continous Integration*) Jenkins. Jest to jeden z najpopularniejszych systemów ciągłej integracji oprogramowania. Filtrując wyniki po adresach URL zdobyto informacje o 1.5 tysiąca serwerów tego typu.

Usługami, które ściśle wiążą się z wspomnianymi wcześniej systemami kontroli wersji, czy serwerami CI łączą się charakterystyczne, testowe systemy. Jeśli chodzi o płaszczyznę przedstawioną w niniejszej pracy magisterskiej, są to najczęściej testowe strony internetowe, których adresy rozpoczynają się od prefiku „.test”. Strony takie często nie są wystarczająco dobrze zabezpieczone, więc dla cyberprzestępca może to być jeden ze sposobów, aby zwiększyć poziom swoich uprawnień w danym systemie.

#### **4.7.2 Usługi poczty elektronicznej**

Z zeskanowanych danych można również pozyskiwać informacje o usługach poczty elektronicznej. Wyróżniono kilka podejść do poszukiwania takich informacji. Jednym z nich jest poszukiwanie adresów domen z prefiksem *owa*. Odpowiadają one najczęściej systemom pocztowym firmy Microsoft (ang. *Outlook Web Access*) i często, dla ułatwienia zapamiętania adresu tej usługi, skracają jej adres do właśnie tego skrótu. Innym podejściem jest pobieranie ze stref DNS rekordów o identyfikatorze równym 15, czyli serwerów wymiany poczty.

W niniejszej pracy magisterskiej zdecydowano się na przeprowadzenie analizy zarówno jedną jak i drugą opisaną metodą. Wyniki zostały zaprezentowane poniżej:

- znaleziono 3378 rekordów, które odnoszą się do serwisów *Outlook Web Access*; są to rekordy typu A, AAAA lub CNAME, więc odnoszą się odpowiednio do adresów IPv4, adresów IPv6

lub są zwykłymi aliasami dla innych domen,

- zostały wyodrębnione 3443626 rekordów typu MX (id 15); oprócz typowego zastosowania rekordu MX, czyli definiowania serwerów poczty elektronicznej wewnątrz domenyauważono, że niektóre ze znalezionych wpisów odnoszą się do usług pocztowych firmy Google.

Specyficzny rekord MX, który odnosi się do usług firmy Google można rozpoznać po jego adresie. Jest najczęściej postaci *alt{id}.aspmx.l.google.com* lub *aspmx{id}.googlemail.com*. Obecność takiego wpisu w strefie DNS sugeruje, że organizacja, która wykorzystuje daną domenę korzysta z usług określanych nazwą *G Suite*. Nazwa odnosi się głównie do usług poczty elektronicznej dla firm udostępnianych przez Google jak również do rozwiązań Cloud – edycji czy przechowywania dokumentów w chmurze.

Oprócz tego, że informacje na temat usług poczty elektronicznej są istotne ze względów czysto infrastrukturalnych, czyli cyberprzestępca pozyskuje informacje na temat uruchomionego serwera poczty, to mogą być również ciekawe ze względu na możliwość użycia ich w atakach phishingowych. Niewątpliwie informacja taka jak typ używanego systemu pocztowego (*Outlook Web Access*, *G Suite* czy inne) są istotną informacją dla takiego ataku i pozwalają lepiej się do niego przygotować.

## 4.8 Strefy DNSSEC

Informacje zebrane podczas analizy danych pobranych przez skaner zostały poddane analizie pod względem liczby serwerów, które wspierają DNSSEC. Stwierdzono, że 16225 stref DNS miało w swoich wpisach typ rekordu RRSIG i na tej podstawie można domniemywać, że serwery obsługujące te domeny mają wdrożone rozwiązanie DNSSEC. Sprawdzono, jak duża różnica występuje pod względem wielkości odebranych pakietów w zależności od włączenia bądź wyłączenia rozszerzenia DNSSEC.

Informacja o tym, że dana strefa wspiera rozszerzenia DNSSEC jest niezwykle istotna w kontekście ataków typu *amplification attack*. Atak ten, w odniesieniu do protokołu DNS, polega na tym, że atakujący wykorzystuje serwery DNS aby przeprowadzić atak DDoS (ang. *Distributed Denial of Service*) na wybrany cel. Atak DDoS polega na generowaniu bardzo dużego ruchu sieciowego do danego serwera, tak, aby inni nie mogli skorzystać z jego usług. Jednym ze sposobów realizacji takiego ataku jest tak zwany *amplification attack*, gdzie atakujący preparuje datagramy UDP ze zmienionym źródłowym adresem IP. Wysyła wiele takich zapytań, na przykład do różnych serwerów DNS, które odpowiadają na podmieniony adres IP, powodując przeciążenia. Rozszerzenie DNSSEC jest dużo lepsze do tego rodzaju ataków, z racji, że oprócz odpowiedzi DNS wysyłany jest podpis danego rekordu. Powołując się na publikację [21], można policzyć zdefiniowany tam współczynnik wzmacnienia (ang. *bandwidth amplification factor*), czyli stosunek długości odpowiedzi do długości zapytania. Policzono współczynnik dla jednej z domen, które zostały zakwalifikowane

jako domeny wykorzystujące DNSSEC.

$$BAF = \frac{\text{len}(UDP)\text{odpowiedzi}}{\text{len}(UDP)\text{zapytania}} = \frac{5544}{632} = 8,71$$

Domeną tą była domena tailz.nl, a adres serwera autorytywnego to 5.200.7.135. Współczynnik ten zależy w dużym stopniu od tego, jaki algorytm został wybrany do podpisywania rekordów. Im podpis jest dłuższy i bezpieczniejszy, tym oczywiście współczynnik amplifikacji jest wyższy.

Sprawdzono także, czy odpytanie domeny o wszystkie rekordy (modyfikator *any* programu dig) pozwoli na zwiększenie tego współczynnika, jednak w przypadku domen DNSSEC często istnieje uzasadniona obawa, że odpowiedź będzie większa, niż maksymalny rozmiar datagramu UDP i będzie wymagana zmiana protokołu warstwy transportu na protokół TCP. W przypadku ataku opisywanego w tym rozdziale wymagane jest, aby używać protokołu UDP aby nie było potrzeby zestawiania sesji oraz w prosty sposób można było manipulować źródłowym adresem IP.

## 4.9 Wpisy SPF

Wpisy SPF (ang. *Sender Policy Framework*) zostały zdefiniowane w celu ochrony użytkowników domeny przed spamem. Pozwalają na określanie reguł, czy wiadomość elektroniczna powinna zostać przyjęta przez serwer czy nie.

Istotnym elementem wpisów SPF są między innymi kwalifikatory. Wyróżniamy kwalifikatory: „+”, „-”, „?”, „”. Szczególnie interesujący jest kwalifikator +all, który mówi, że reguły SPF pozwalały na odebranie wszystkich wiadomości ze wszystkich domen. Wśród danych pobranych podczas skanowania znaleziono 1.8 miliona wpisów SPF. Co ciekawsze, 3.5 tysiąca z nich było opatrzonych klauzulą „+all”. Daje to między innymi bardzo ciekawą informację dla osób wysyłających niechcianą pocztę (tzw. spam), czy podmiotów, które rozsyłają wiadomości z linkami do złośliwego oprogramowania (ang. *phishing*). Wiedząc, że reguły SPF nie odfiltrują danej wiadomości warto jest próbować wysyłać ją do użytkowników mających adres w danych domenach. Cyberprzestępca może wtedy przypuszczać, że jego spam rzeczywiście trafi do odbiorcy oraz oszczędza na czasie, nie wysyłając wiadomości do serwerów, które najprawdopodobniej jego spam odfiltrują. Istotne jest, że rekordy SPF są enkapsulowane w rekordach DNS o numerze identyfikacyjnym 16 (TXT), gdzie mogą znajdować się również inne informacje, które są niezwiązane z mechanizmem SPF. W związku z tym nie ma skutecznej metody na pobieranie tylko i wyłącznie rekordów SPF. Przykłady wpisu definiującego politykę wysyłania/odbierania wiadomości przedstawiono na listingu 4.5.

- 
- 1 gfyginfo.writerscollective.org. 14400 16 v=spf1 +a +mx +ip4 $\leftarrow$   
:209.200.229.240 ~all
  - 2 goodforyougoodies.writerscollective.org. 14400 16 v=spf1 +a +mx + $\leftarrow$   
ip4:209.200.229.240 ~all
  - 3 wrallp.com. 300 16 v=spf1 mx ip4:209.124.166.197 ip4:209.166.136.200 $\leftarrow$

```
ip4:4.35.225.147 -all
4 bexc3.ad.wrallp.com. 300 16 v=spf1 a -all
5 woonbetonmeubels.nl. 180 16 v=spf1 +a +mx +all
```

---

Listing 4.5: Przykład odebranych rekordów SPF.

Informacje, jakie można odczytać z podanych rekordów SPF, to na przykład linijka 1. listingu 4.5 definiuje, że jeśli wiadomość pochodzi z innego systemu niż dostępny pod adresem ip 209.200.229.240, domena z której wysłano wiadomość nie posiada wpisu A bądź MX, to należy wiadomość otagować. Linia 3. definiuje regułę, która pozwala na przyjmowanie wiadomości jedynie od trzech systemów, zaś reguła z linii 5. pozwala na odebranie każdej wiadomości.

Mając informacje o tym, jakie wiadomości nie są odrzucane od danych serwerów pocztowych, atakujący może lepiej zaplanować swój atak. Jeśli wie, że dany serwer na pewno nie otrzyma wiadomości od niego, to albo nie będzie próbował jej wysyłać, albo będzie się starał oszukać system, preparując wiadomość, która będzie pasowała do reguł zdefiniowanych w domenie. Ma to szcze- gólne znaczenie w momencie, gdy atakującemu zależy na tym, aby wiadomość trafiła dokładnie do odbiorcy w danej strefie. Pewność ta jest ważna w przypadku testowania, ataków na konkretne domeny, które najczęściej odpowiadają konkretnym instytucjom czy firmom.

## 4.10 Strefy podsieci

Podczas realizacji pracy magisterskiej zauważono w zebranych danych obecność specyficznych domen, których rozmiar wynosił od 250 do 260 wpisów. Zyskały one zainteresowanie dlatego, że znalazły się wysoko w zestawieniu liczebności domen o danej liczbie wpisów w pliku strefy. Okazuje się, że również ich rozmiar nie jest przypadkowy. Domeny te najczęściej składają się z wpisów, które mapują podsieci wykorzystujące adresy IP klasy C na nazwy domenowe. Przykład takiego pliku strefy został zaprezentowany na listingu 4.6.

---

```
1 5181115.cn. 86400 6 dns1.zhangcong.top.5181115.cn. root.5181115.cn. ←
   1852269423 1852244852 1869660172 74608495 1958743040
2 5181115.cn. 86400 2 dns1.zhangcong.top
3 5181115.cn. 86400 2 dns2.zhangcong.top
4 *.5181115.cn. 86400 1 107.163.236.2
5 *.5181115.cn. 86400 1 107.163.236.3
6 (... )
7 *.5181115.cn. 86400 1 107.163.236.253
8 *.5181115.cn. 86400 1 107.163.236.254
9 dns1.5181115.cn. 86400 1 115.29.128.246
10 dns2.5181115.cn. 86400 1 139.129.18.2
11 5181115.cn. 86400 6 dns1.zhangcong.top.5181115.cn. root.5181115.cn. ←
```

```
1 86400 3600 604800 10800
```

---

Listing 4.6: Przykład domeny mapującej podsieć.

W przypadku zaprezentowanym na listingu 4.6 widać, że każda z nazw domenowych może być tłumaczona na około 250 adresów IP. Podobne przypadki zaobserwowano dla niektórych, lokalnych firm świadczących usługi dostępu do internetu

Sprawdzono, czy możliwe jest wykorzystanie stref tego typu w ataku typu *amplification attack*. Niestety, pomimo że strefa zawiera wiele wpisów, modyfikator *any* programu dig spowodował odebranie jedynie dwóch rekordów DNS, a współczynnik amplifikacji takiej komunikacji wynosił odpowiednio:

$$BAF = \frac{\text{len}(UDP)\text{odpowiedź}}{\text{len}(UDP)\text{zapytania}} = \frac{1264}{664} = 1,90$$

Okazuje się, że jest dużo niższy niż współczynnik dla wzmacnienia przy użyciu serwerów DNSSEC a ponadto, dużych stref jest znacznie mniej niż serwerów, które używają protokołu DNS-SEC.



# 5. Podsumowanie

W niniejszej pracy magisterskiej zaproponowano unikatowe podejście do problemu transferu strefy AXFR oparte na globalnym skanowaniu domen i ich serwerów przestrzeni nazw. Przeanalizowano, czy powszechnie dostępne narzędzia zapewniają takie funkcje, aby przeprowadzić badania w zakresie postawionego problemu. Została zaproponowana implementacja oraz wdrożenie systemu, który umożliwia optymalne wykonanie takich badań. Zebrano oraz przeanalizowano dane, które można uzyskać w wyniku globalnego skanowania domen pod kątem podatności na transfer AXFR. Pogrupowano uzyskane odpowiedzi, typy zachowań na wystosowane żądanie AXFR. Opisano, w jaki sposób mogą odpowiadać serwery autorytatywne oraz przeanalizowano popularność każdego z typów otrzymywanej odpowiedzi.

Podsumowując przeprowadzone prace i badania okazuje się, że transfer strefy wykorzystując mechanizm AXFR jest wciąż możliwy dla bardzo wielu domen. Bardzo duża część domen jest pod zarządem niewielkiej liczby serwerów autorytatywnych, więc możliwe, że zezwolenie na taki transfer bywa zamierzone. Resztę można zaliczyć jako niepoprawnie skonfigurowane i to one są najbardziej narażone na ataki cyberprzestępco. Zalecenia wydane w dalszej części rozdziału są kierowane głównie w kierunku administratorów właśnie tych domen.

W pracy opisano także, które części stref DNS pozwalają cyberprzestępcom na pozyskanie informacji na temat ich potencjalnych celów. Skupiono się w głównej mierze na usługach uruchamianych w badanych systemach, informacjach na temat infrastruktury antyspamowej, ale także na pośrednim wykorzystaniu danych maszyn w innych typach ataków.

## 5.1 Zalecenia

Transfer strefy AXFR pozwala na uzyskanie wielu informacji na temat tego, jakie usługi oraz jakie maszyny są uruchomione wewnętrz danej domeny. Drastycznym, ale najbardziej skutecznym zaleceniem jest wycofanie z implementacji serwerów DNS funkcji transferu strefy DNS do każdej maszyny wystosowującej zapytanie tego typu. Wymagałoby to używania jedynie adresów IP serwerów, które mogą ten transfer przeprowadzić. Rozwiążanie to ograniczyłoby skalowalność rozwiązania, jednak jawne podanie adresu IP jest dużo bardziej bezpieczne i zapewnia, że transfer strefy przeprowadzony będzie tylko przez te maszyny, którym rzeczywiście, wprost, na to zezwolono.

Innym rozwiązaniem jest rezygnacja z własnych serwerów DNS na rzecz usług *DNS as a service* (rozdział 2.4). Takie postępowanie eliminuje większość problemów związanych z transferem strefy, gdyż cała infrastruktura jest dzierżawiona od zewnętrznej firmy. Firmom takim również zależy na jak najlepszej jakości ich usług oraz zależy im, aby dane na temat stref DNS nie były tak powszechnie dostępne, dlatego dobrze zabezpieczają transfer strefy. Odbywa się to na przykład dzięki zestawieniu sesji SSL pomiędzy stronami wymieniającymi dane, bądź innym wykorzystaniu kryptografii (na przykład szyfrowanie).

## 5.2 Retrospekcja

Pośrednim wynikiem tworzenia niniejszej pracy magisterskiej mogą być również wnioski na temat samego procesu jej realizacji czy przeprowadzania badań. Nie wszystkie etapy pracy były przeprowadzane płynnie, co pozwala na wyciągnięcie wniosków pozwalających uniknąć takich sytuacji w przyszłości. Jednym z trudniejszych etapów realizacji pracy było przygotowanie oprogramowania i środowiska pod globalne skanowanie. Trud ten wynika głównie ze skali, skrypty dostępne w Internecie nie są aż tak wydajne aby zapewnić oczekiwana wydajność. Poważnym błędem, który spowodował wydłużenie czasu uruchomienia systemu były próby wykorzystania już istniejących narzędzi, jednak pozwoliło to na opisanie ich wad bądź zalet. Dodatkowo, można wyciągnąć wniosek, że przeprowadzanie skanowania na szeroką skalę bardzo często wymaga implementacji oprogramowania dokładnie pod kątem tego specyficznego przypadku, bardzo często bez używania innych projektów powiązanych z tematyką badań. Oprogramowanie skanera jest ciągle ulepszane, co dowodzi temu, że nawet kilka miesięcy działania systemu nie pozwala na pełne, dokładne przetestowanie skanera.

# Bibliografia

- [1] Netcat 1.10, 2007. <http://nc110.sourceforge.net>.
- [2] Dns zone transfer attack, 2012. <https://security.stackexchange.com/questions/10452/dns-zone-transfer-attack>.
- [3] Ixfr confuses dig, godaddy, 2013.
- [4] Ixfr response must be soa only when request serial >= current serial, 2014. <https://github.com/PowerDNS/pdns/issues/1192>.
- [5] Alexa internet, 2017. <http://www.alexa.com>.
- [6] Team cymru research nfp, 2017. <http://www.team-cymru.org>.
- [7] D. Eastlake 3rd. Secure domain name system dynamic update. RFC 2137, RFC Editor, April 1997.
- [8] D. Eastlake 3rd. Secret key establishment for dns (tkey rr). RFC 2930, RFC Editor, September 2000.
- [9] D. Eastlake 3rd. Hmac sha (hashed message authentication code, secure hash algorithm) tsig algorithm identifiers. RFC 4635, RFC Editor, August 2006.
- [10] Donald E. Eastlake 3rd and Charles W. Kaufman. Domain name system security extensions. RFC 2065, RFC Editor, January 1997. <http://www.rfc-editor.org/rfc/rfc2065.txt>.
- [11] Brian Aitken. Interconnect communication mc / 080:dnssec deployment study, 2011. <http://stakeholders.ofcom.org.uk/binaries/internet/domain-name-security.pdf>.
- [12] Ken Silva Cricket Liu Allen Householder, Brian King. Securing an internet name server, 2002. [https://resources.sei.cmu.edu/asset\\_files/WhitePaper/2002\\_019\\_001\\_52496.pdf](https://resources.sei.cmu.edu/asset_files/WhitePaper/2002_019_001_52496.pdf).

- [13] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Dns security introduction and requirements. RFC 4033, RFC Editor, March 2005. <http://www.rfc-editor.org/rfc/rfc4033.txt>.
- [14] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol modifications for the dns security extensions. RFC 4035, RFC Editor, March 2005. <http://www.rfc-editor.org/rfc/rfc4035.txt>.
- [15] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource records for the dns security extensions. RFC 4034, RFC Editor, March 2005. <http://www.rfc-editor.org/rfc/rfc4034.txt>.
- [16] Internet Assigned Numbers Authority. Technical requirements for authoritative name servers, 2017. <https://www.iana.org/help/nameserver-requirements>.
- [17] PowerDNS.COM B.V. Powerdns, 2017. <https://www.powerdns.com>.
- [18] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer. Openpgp message format. RFC 4880, RFC Editor, November 2007. <http://www.rfc-editor.org/rfc/rfc4880.txt>.
- [19] Team Censys. Forward dns (fdns any), 2016. [https://scans.io/study/sonar.fdns\\_v2](https://scans.io/study/sonar.fdns_v2).
- [20] US Cert. Dns zone transfer axfr requests may leak domain information, 2015. <https://www.us-cert.gov/ncas/alerts/TA15-103A>.
- [21] Horst Gortz Christian Rossow. Amplification hell: Revisiting network protocols for ddos abuse, 2014. [https://www.internetsociety.org/sites/default/files/01\\_5.pdf](https://www.internetsociety.org/sites/default/files/01_5.pdf).
- [22] Inc. Cloudflare. Cloudflare's dns firewall, 2017. <https://www.cloudflare.com/dns/dns-firewall/>.
- [23] L. Daigle. Whois protocol specification. RFC 3912, RFC Editor, September 2004.
- [24] Vacha Dave, Saikat Guha, and Yin Zhang. Measuring and fingerprinting click-spam in ad networks. In *Proceedings of the Special Interest Group on Data Communication (SIGCOMM)*, Helsinki, Finland, August 2012.
- [25] TENABLE NETWORK SECURITY INC. Dave Breslin. Ssh server vulnerabilities, 2012. <http://static.tenable.com/oldsite/blog/files/sample---ssh-server-vulnerabilities.pdf>.

- [26] Christopher Davis, Paul Vixie, Tim Goodwin, and Ian Dickinson. A means for expressing location information in the domain name system. RFC 1876, RFC Editor, January 1996. <http://www.rfc-editor.org/rfc/rfc1876.txt>.
- [27] Shumon Huque Moni Naor Jan Vcelak Leonid Rezyin Sharon Goldberg Dimitrios Papadopoulos, Duane Wessels. Making nsec5 practical for dnssec, 2017. <https://eprint.iacr.org/2017/099.pdf>.
- [28] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A search engine backed by Internet-wide scanning. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security*, October 2015.
- [29] D. Eastlake and T. Hansen. Us secure hash algorithms (sha and hmac-sha). RFC 4634, RFC Editor, July 2006. <http://www.rfc-editor.org/rfc/rfc4634.txt>.
- [30] D. Eastlake and P. Jones. Us secure hash algorithm 1 (sha1). RFC 3174, RFC Editor, September 2001. <http://www.rfc-editor.org/rfc/rfc3174.txt>.
- [31] Delft University of Technology Economics of Cybersecurity research group. pyasn 1.6.0b1 offline ip address to autonomous system number lookup module, 2017. <https://pypi.python.org/pypi/pyasn>.
- [32] Paul Eggert. shuf make random permutation, 2017. <https://linux.die.net/man/1/shuf>.
- [33] C.F. Everhart, L.A. Mamakos, R. Ullmann, and P.V. Mockapetris. New dns rr definitions. RFC 1183, RFC Editor, October 1990.
- [34] Policy Faculty of Technology and Management (TPM) at Delft University of Technology. Economics of cybersecurity group. <https://www.tudelft.nl/>.
- [35] P. Faltstrom and M. Mealling. The e.164 to uniform resource identifiers (uri) dynamic delegation discovery system (ddds) application (enum). RFC 3761, RFC Editor, April 2004.
- [36] International Organization for Standardization. Programming languages — C (C11). Standard, International Organization for Standardization, April 2011.
- [37] Python Software Foundation. *Python Language Reference, version 2.7*. Python Software Foundation., 2017.
- [38] Pierre-Alain Fouque, Gaëtan Leurent, and Phong Q. Nguyen. Full key-recovery attacks on hmac/nmac-md4 and nmac-md5. <https://www.di.ens.fr/~fouque/pub/crypto07b.pdf>.

- [39] Google. Google ipv6 statistics, 2017. <https://www.google.com/intl/en/ipv6/statistics.html>.
- [40] Arnt Gulbrandsen, Paul Vixie, and Levon Esibov. A dns rr for specifying the location of services (dns srv). RFC 2782, RFC Editor, February 2000. <http://www.rfc-editor.org/rfc/rfc2782.txt>.
- [41] M. S. Dahiya H. P. Sanghvi. Cyber reconnaissance: An alarm before cyber attack, 2013. <http://research.ijcaonline.org/volume63/number6/pxc3885202.pdf>.
- [42] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an autonomous system (as). BCP 6, RFC Editor, March 1996.
- [43] Stephen Haywood. axfr, 2017. <https://github.com/averagesecurityguy/axfr>.
- [44] R. Hinden and S. Deering. Ip version 6 addressing architecture. RFC 4291, RFC Editor, February 2006. <http://www.rfc-editor.org/rfc/rfc4291.txt>.
- [45] SEDO HOLDING. Sedo holding ag 6-month report, 2017. [https://www.united-internet.de/uploads/tx\\_unitedinternetpublication/United\\_Internet\\_q22017e\\_02.pdf](https://www.united-internet.de/uploads/tx_unitedinternetpublication/United_Internet_q22017e_02.pdf).
- [46] Kim Hubbard, Mark Kosters, David Conrad, Daniel Karrenberg, and Jon Postel. Internet registry ip allocation guidelines. BCP 12, RFC Editor, November 1996. <http://www.rfc-editor.org/rfc/rfc2050.txt>.
- [47] Imperva. Incapsula name server (ns) protection, 2017. <https://www.incapsula.com/dns-ddos-protection-services.html>.
- [48] Google Inc. Google domains help, 2017. <https://support.google.com/domains/>.
- [49] NASK instytut badawczy. Podpisy tsig, 2017. [https://www.dns.pl/dnssec/theory\\_tsig.html](https://www.dns.pl/dnssec/theory_tsig.html).
- [50] International Seismological Centre. *BIND The most widely used Name Server Software*. Internat'l. Seismol. Cent., Thatcham, United Kingdom, 2017. <http://www.isc.org>.
- [51] International Seismological Centre. Dig dns lookup utility, 2017. <https://linux.die.net/man/1/dig>.
- [52] Brian W. Kernighan. The c programming language. In Dennis M. Ritchie, editor, *The C Programming Language*. Prentice Hall Professional Technical Reference, 2nd edition, 1988.

- [53] S. Kwan, P. Garg, J. Gilroy, L. Esibov, J. Westhead, and R. Hall. Generic security service algorithm for secret key transaction authentication for dns (gss-tsig). RFC 3645, RFC Editor, October 2003.
- [54] B. Laurie, G. Sisson, R. Arends, and D. Blacka. Dns security (dnssec) hashed authenticated denial of existence. RFC 5155, RFC Editor, March 2008. <http://www.rfc-editor.org/rfc/rfc5155.txt>.
- [55] E. Lewis and A. Hoenes. Dns zone transfer protocol (axfr). RFC 5936, RFC Editor, June 2010.
- [56] Nexusguard Limited. Dns protection, 2017.
- [57] J. Linn. Generic security service application program interface version 2, update 1. RFC 2743, RFC Editor, January 2000.
- [58] Cricket Liu and Paul Albitz. *DNS and BIND (5th Edition)*. O'Reilly Media, Inc., 2006.
- [59] Robtex LTD. Robotex dns, 2017. <https://www.robtex.com>.
- [60] Teluu LTD. Pjlib – open source, small footprint framework library, 2017. <http://www.pjsip.org>.
- [61] Michel van Eeten Maciej Korczyński, Michał Król. Zone poisoning: The how and where of non-secure dns dynamic updates, 2016. <http://mkorczynski.com/IMC16Korczynski.pdf>.
- [62] M. Mealling and R. Daniel. The naming authority pointer (naptr) dns resource record. RFC 2915, RFC Editor, September 2000.
- [63] Steven M. Mellovin. Using the domain name system for system break-ins, 1995. <https://www.cs.columbia.edu/~smb/papers/dnshack.pdf>.
- [64] Microsoft. How dns works, 2017. [https://technet.microsoft.com/en-us/library/cc772774\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc772774(v=ws.10).aspx).
- [65] D. Mills, J. Martin, J. Burbank, and W. Kasch. Network time protocol version 4: Protocol and algorithms specification. RFC 5905, RFC Editor, June 2010. <http://www.rfc-editor.org/rfc/rfc5905.txt>.
- [66] P. Mockapetris. Domain names - concepts and facilities. STD 13, RFC Editor, November 1987. <http://www.rfc-editor.org/rfc/rfc1034.txt>.
- [67] P. Mockapetris. Domain names - implementation and specification. STD 13, RFC Editor, November 1987. <http://www.rfc-editor.org/rfc/rfc1035.txt>.

- [68] NASK. Cert polska, 2017. <https://www.cert.pl>.
- [69] Sebastian Neef. Axfr-test, 2016. <https://github.com/internetwache/Python-AXFR-Test>.
- [70] Masataka Ohta. Incremental zone transfer in dns. RFC 1995, RFC Editor, August 1996. <http://www.rfc-editor.org/rfc/rfc1995.txt>.
- [71] Oracle. System administration guide: Naming and directory services, 2017. <https://docs.oracle.com/cd/E19683-01/817-4843/dnsintro-70/index.html>.
- [72] O'Reilly and Associates. Dns and bind fourth edition, 2002. [https://docstore.mik.ua/orelly/networking\\_2ndEd/dns/ch02\\_04.htm](https://docstore.mik.ua/orelly/networking_2ndEd/dns/ch02_04.htm).
- [73] Yakov Rekhter, Robert G. Moskowitz, Daniel Karrenberg, Geert Jan de Groot, and Eliot Lear. Address allocation for private internets. BCP 5, RFC Editor, February 1996. <http://www.rfc-editor.org/rfc/rfc1918.txt>.
- [74] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol. RFC 3261, RFC Editor, June 2002. <http://www.rfc-editor.org/rfc/rfc3261.txt>.
- [75] P. Saint-Andre. Extensible messaging and presence protocol (xmpp): Instant messaging and presence. RFC 3921, RFC Editor, October 2004.
- [76] Marcos Sanz. Dnssec and the zone enumeration, 2004. [http://www.denic.de/fileadmin/public/events/DNSSEC\\_testbed/zone-enumeration.pdf](http://www.denic.de/fileadmin/public/events/DNSSEC_testbed/zone-enumeration.pdf).
- [77] J. Schlyter. Dns security (dnssec) nextsecure (nsec) rdata format. RFC 3845, RFC Editor, August 2004.
- [78] Farsight Security. Dns database (dns-db), 2017. <https://www.dnsdb.info>.
- [79] Paula Greve Shane Keats, Dan Nunes. Mapping the mal web, the world's riskiest domains, 2007. [https://promos.mcafee.com/en-US/PDF/Mapping\\_Mal\\_Web.pdf](https://promos.mcafee.com/en-US/PDF/Mapping_Mal_Web.pdf).
- [80] Paula Greve Shane Keats, Dan Nunes. Mapping the mal web, the world's riskiest domains, 2010. [https://promos.mcafee.com/en-US/PDF/MTMW\\_Report.pdf](https://promos.mcafee.com/en-US/PDF/MTMW_Report.pdf).
- [81] Davey Song. An ixfr fallback to axfr case. Internet-Draft draft-song-dnsop-ixfr-fallback-00, IETF Secretariat, May 2016. <http://www.ietf.org/internet-drafts/draft-song-dnsop-ixfr-fallback-00.txt>.
- [82] Eihal Alowaisheq Zhou Li XiaoFeng Wang Sumayah Alrwais, Kan Yuan. Understanding the dark side of domain parking, 2014. <https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-alrwais.pdf>.

- [83] Censys Team. Internet-wide scan data repository, 2015. <https://scans.io/study/hanno-axfr>.
- [84] Akamai Technologies. Dns services and security, 2017. <https://www.akamai.com/us/en/solutions/why-akamai/dns-services.jsp>.
- [85] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi. Dns extensions to support ip version 6. RFC 3596, RFC Editor, October 2003.
- [86] Geoff Huston Tony Bates, Philip Smith. Cidr report, 2017. <https://www.cidr-report.org/as2.0/>.
- [87] Ed Warnicke Ulf Lamping, Richard Sharpe. Wireshark 2.1, 2017. .
- [88] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington. Secret key transaction authentication for dns (tsig). RFC 2845, RFC Editor, May 2000.
- [89] Paul Vixie. A mechanism for prompt notification of zone changes (dns notify). RFC 1996, RFC Editor, August 1996. <http://www.rfc-editor.org/rfc/rfc1996.txt>.
- [90] Paul Vixie and Vernon Schryver. Dns response policy zones (rpz). Internet-Draft draft-ietf-dnsop-dns-rpz-00, IETF Secretariat, March 2017. <http://www.ietf.org/internet-drafts/draft-ietf-dnsop-dns-rpz-00.txt>.
- [91] W3Techs. Usage of top level domains for websites, 2017. [https://w3techs.com/technologies/overview/top\\_level\\_domain/all](https://w3techs.com/technologies/overview/top_level_domain/all).
- [92] Xiaoyun Wang, Hongbo Yu, Wei Wang, Haina Zhang, and Tao Zhan. Cryptanalysis on hmac/hmac-md5 and md5-mac. <https://pdfs.semanticscholar.org/1cc5/63b1eca6695351783e97711b4455c1eca851.pdf>.
- [93] M. Wong and W. Schlitt. Sender policy framework (spf) for authorizing use of domains in e-mail, version 1. RFC 4408, RFC Editor, April 2006.
- [94] Robin Wood. Zonetransfer.me, 2017. <https://digi.ninja/projects/zonetransferme.php>.



# Spis rysunków

1.1	Herarchia systemu DNS [71]. . . . .	10
1.2	Położenie domeny w przestrzeni nazw [48]. . . . .	12
1.3	Zakres drzewa hierarchii DNS wchodzący w skład domeny [72]. . . . .	13
1.4	Zakres drzewa DNS określany terminem strefy DNS [64]. . . . .	14
2.1	Przepływ informacji podczas odpytywania serwera przestrzeni nazw o domenę. . . . .	31
3.1	Algorytm dynamicznego ustawiania limitu czasu żądania dla gniazda TCP w blokującym trybie pracy. . . . .	41
3.2	Algorytm postępowania w przypadku odebrania APDU protokołu DNS. . . . .	43
3.3	Algorytm mieszania linii w dużym pliku wejściowym. . . . .	45
4.1	Wykres liczebności poszczególnych zbiorów typów odpowiedzi. . . . .	50
4.2	Zasada działania narzędzia mapującego adresy IP na numery AS. . . . .	53
4.3	Pochodzenie geograficzne adresów IPv4 z podziałem na państwa. Adresy określane na podstawie numerów AS. . . . .	54
4.4	Zestawienie systemów autonomicznych, z których domeny najczęściej odpowiadają na zapytania AXFR. . . . .	54
4.5	Pochodzenie geograficzne adresów IPv6 z podziałem na państwa. Adresy określane na podstawie numerów AS. . . . .	55
4.6	Pochodzenie geograficzne adresów IPv6 z podziałem na państwa. Adresy określane na podstawie numerów AS. . . . .	56
4.7	Popularność TLD w internecie (dostęp na dzień 15. maja 2017), źródło: [91]. . . . .	57
4.8	Popularność TLD w odpowiedziach AXFR. . . . .	58
4.9	Zestawienie odpowiedzi ze względu na rozkład danych wejściowych. Rozkład odpowiedzi znaleziono do całkowitej liczby odpowiedzi, zaś rozkład zapytań do całkowitej ich liczby. . . . .	59
4.10	Zestawienie częstości występowania domen najwyższego poziomu dla domen, dla których możliwy jest AXFR z ich populacją w zbiorze danych wejściowych. Rozkład odpowiedzi znaleziono do całkowitej liczby odpowiedzi, zaś rozkład zapytań do całkowitej ich liczby. . . . .	59

4.11	Dystrybuanta domeny najwyższego poziomu w odpytywanych domenach. . . . .	60
4.12	Dystrybuanta domeny najwyższego poziomu w zbiorze domen odpowiadających strefą DNS na zapytanie AXFR. . . . .	61
4.13	Rozkład liczby rekordów w odpowiedziach AXFR. . . . .	62
4.14	Rozkład liczby rekordów w odpowiedziach AXFR, widok powiększony. . . . .	63
4.15	Przebieg komunikacji z serwerem wysyłającym pakiet TCP z flagą RST. . . . .	66
4.16	Lokalizacja serwerów nazw odpowiadających pakietem TCP z flagą RST. . . . .	67
4.17	TLD domen, których serwery autorytatywne odpowiadają TCP RST. . . . .	68
4.18	Lokalizacja serwerów DNS, które odpowidały więcej niż jednym rekordem. . . . .	69
4.19	Dystrybuanta rozkładu popularności serwerów przestrzeni nazw, które odpowiadają na zapytania AXFR. . . . .	69

# Spis tabel

1.1	Typy rekonesansu, opis na podstawie [41]. . . . .	8
1.2	Typy rekonesansu, opis na podstawie [41]. . . . .	12
1.3	Rodzaje rekordów w bazach danych serwerów przestrzeni nazw. Opis na podstawie [67]. . . . .	17
1.4	Rodzaje rekordów w bazach danych serwerów przestrzeni nazw określonych w rozszerzeniach dokumentu RFC 1035 [67]. . . . .	18
3.1	Podsumowanie istniejących narzędzi. . . . .	38
3.2	Rekordy DNS obsługiwane w skanerze wraz z identyfikatorami. . . . .	42
4.1	Opis zbiorów zapytań/odpowiedzi . . . . .	61
4.2	Tabela liczebności AS w opisywanym przypadku . . . . .	67
4.3	Autorytatywne NS dla największej liczby domen. . . . .	70
4.4	Kwantyle rozkładu popularności serwerów przestrzeni nazw, które odpowiadają na AXFR. . . . .	70